



npp
Release 13.4

NVIDIA Corporation

Jul 15, 2026

Contents:

1	Indices and Search	1
1.1	What is NPP ?	1
1.1.1	Files	1
1.1.1.1	Header Files	2
1.1.1.2	Library Files	2
1.1.2	Library Organization	3
1.1.3	Supported NVIDIA Hardware	5
1.2	General Conventions	5
1.2.1	Memory Management	5
1.2.1.1	Scratch Buffer and Host Pointer	6
1.2.2	Function Naming	7
1.2.3	Integer Result Scaling	7
1.2.4	Rounding Modes	8
1.2.4.1	Rounding Mode Parameter	8
1.3	Image Processing Conventions	8
1.3.1	Function Naming	8
1.3.2	Image Data	9
1.3.2.1	Line Step	9
1.3.2.2	Parameter Names for Image Data	10
1.3.2.3	Image Data Alignment Requirements	14
1.3.2.4	Image Data Related Error Codes	14
1.3.3	Region-Of-Interest (ROI)	14
1.3.3.1	ROI Related Error Codes	15
1.3.4	Masked Operation	15
1.3.5	Channel-of-Interest API	15
1.3.5.1	Select-Channel Source-Image Pointer	16
1.3.5.2	Select-Channel Source-Image	16
1.3.5.3	Select-Channel Destination-Image Pointer	16
1.3.6	Source-Image Sampling	16
1.3.6.1	Point-Wise Operations	17
1.3.6.2	Neighborhood Operations	17
1.4	Signal Processing Conventions	18
1.4.1	Signal Data	18
1.4.1.1	Parameter Names for Signal Data	19
1.4.1.2	Signal Data Alignment Requirements	19
1.4.1.3	Signal Data Related Error Codes	20
1.4.2	Signal Length	20
1.4.2.1	Length Related Error Codes	20
1.5	Data Types, Structs, Enums, and Constants	20
1.5.1	Application Managed Stream Context	47
1.6	Core NPP Functions	52
1.7	Image Arithmetic And Logical Operations	52
1.7.1	Arithmetic Operations	52

1.7.1.1	Arithmetic Operations	52
1.7.1.2	AddC	52
1.7.1.3	MulC	93
1.7.1.4	MulCScale	134
1.7.1.5	SubC	146
1.7.1.6	DivC	187
1.7.1.7	AbsDiffC	229
1.7.1.8	Add	231
1.7.1.9	AddSquare	258
1.7.1.10	AddProduct	261
1.7.1.11	AddWeighted	264
1.7.1.12	Mul	267
1.7.1.13	MulScale	293
1.7.1.14	Sub	300
1.7.1.15	Div	327
1.7.1.16	Div_Round	354
1.7.1.17	Abs	366
1.7.1.18	AbsDiff	373
1.7.1.19	Sqr	376
1.7.1.20	Sqrt	389
1.7.1.21	Ln	400
1.7.1.22	Exp	407
1.7.2	Logical Operations	412
1.7.2.1	Logical Operations	412
1.7.2.2	AndC	413
1.7.2.3	OrC	421
1.7.2.4	XorC	430
1.7.2.5	RShiftC	438
1.7.2.6	LShiftC	452
1.7.2.7	And	461
1.7.2.8	Or	470
1.7.2.9	Xor	479
1.7.2.10	Not	489
1.7.3	Image Alpha Composition Operations	492
1.7.3.1	AlphaCompC	492
1.7.3.2	AlphaComp	505
1.8	Image Color Conversion Functions	512
1.8.1	Color Processing Functions	512
1.8.1.1	Color To Gray Conversion	536
1.8.1.2	Color Debayer	546
1.8.1.3	Color Gamma Correction	550
1.8.1.4	Complement Color Key	554
1.8.1.5	ColorTwist	556
1.8.1.6	ColorTwistBatch	587
1.8.1.7	ColorLUT	602
1.8.1.8	ColorLUTLinear	619
1.8.1.9	ColorLUTCubic	638
1.8.1.10	ColorLUTTrilinear	656
1.8.1.11	ColorLUTPalette	659
1.8.2	Color Sampling Format Conversion Functions	667
1.8.2.1	YCbCr420ToYCbCr411	667
1.8.2.2	YCbCr422ToYCbCr422	668
1.8.2.3	YCbCr422ToYCrCb422	670
1.8.2.4	YCbCr422ToCbYCr422	671

1.8.2.5	CbYCr422ToYCbCr411	671
1.8.2.6	YCbCr422ToYCbCr420	672
1.8.2.7	YCrCb420ToYCbCr422	674
1.8.2.8	YCbCr422ToYCrCb420	675
1.8.2.9	YCbCr422ToYCbCr411	675
1.8.2.10	YCrCb422ToYCbCr422	677
1.8.2.11	YCrCb422ToYCbCr420	678
1.8.2.12	YCrCb422ToYCbCr411	678
1.8.2.13	CbYCr422ToYCbCr422	679
1.8.2.14	CbYCr422ToYCbCr420	680
1.8.2.15	CbYCr422ToYCrCb420	681
1.8.2.16	YCbCr420ToYCbCr420	681
1.8.2.17	YCbCr420ToYCbCr422	683
1.8.2.18	YCbCr420ToCbYCr422	684
1.8.2.19	YCbCr420ToYCrCb420	685
1.8.2.20	YCrCb420ToCbYCr422	685
1.8.2.21	YCrCb420ToYCbYCr420	686
1.8.2.22	YCrCb420ToYCbYCr411	687
1.8.2.23	YCbCr411ToYCbCr411	687
1.8.2.24	YCbCr411ToYCbCr422	688
1.8.2.25	YCbCr411ToYCrCb422	690
1.8.2.26	YCbCr411ToYCbCr420	691
1.8.2.27	YCbCr411ToYCrCb420	693
1.8.2.28	NV12ToYUV420	693
1.8.3	Color Model Conversion Functions	694
1.8.3.1	RGBToYUV	694
1.8.3.2	BGRToYUV	696
1.8.3.3	YUVToRGB	699
1.8.3.4	YUVToRGBBatch	701
1.8.3.5	YUVToRGBBatchAdvanced	702
1.8.3.6	YUVToBGR	703
1.8.3.7	YUVToBGRBatch	705
1.8.3.8	YUVToBGRBatchAdvanced	706
1.8.3.9	RGBToYUV422	708
1.8.3.10	YUV422ToRGB	709
1.8.3.11	YUV422ToRGBBatch	711
1.8.3.12	YUV422ToRGBBatchAdvanced	711
1.8.3.13	YUV422ToBGRBatch	712
1.8.3.14	YUV422ToBGRBatchAdvanced	713
1.8.3.15	RGBToYUV420	714
1.8.3.16	YUV420ToRGB	715
1.8.3.17	YUV420ToRGBBatch	716
1.8.3.18	YUV420ToRGBBatchAdvanced	717
1.8.3.19	NV12ToRGB	718
1.8.3.20	NV21ToRGB	720
1.8.3.21	BGRToYUV420	720
1.8.3.22	YUV420ToBGR	721
1.8.3.23	YUV420ToBGRBatch	722
1.8.3.24	YUV420ToBGRBatchAdvanced	722
1.8.3.25	NV12ToBGR	723
1.8.3.26	NV21ToBGR	725
1.8.3.27	RGBToYCbCr	725
1.8.3.28	YCbCrToRGB	728
1.8.3.29	YCbCrToRGBBatch	730

1.8.3.30	YCbCrToRGBBatchAdvanced	731
1.8.3.31	YCbCrToBGR	732
1.8.3.32	YCbCrToBGRBatch	733
1.8.3.33	YCbCrToBGRBatchAdvanced	734
1.8.3.34	YCbCrToBGR709CSC	736
1.8.3.35	RGBToYCbCr422	737
1.8.3.36	YCbCr422ToRGB	738
1.8.3.37	YCbCr422ToRGBBatch	739
1.8.3.38	YCbCr422ToRGBBatchAdvanced	740
1.8.3.39	RGBToYCrCb422	741
1.8.3.40	YCrCb422ToRGB	742
1.8.3.41	YCbCr422ToBGR	743
1.8.3.42	YCbCr422ToBGRBatch	744
1.8.3.43	YCbCr422ToBGRBatchAdvanced	745
1.8.3.44	RGBToCbYCr422	746
1.8.3.45	CbYCr422ToRGB	747
1.8.3.46	BGRToCbYCr422	747
1.8.3.47	BGRToCbYCr422 709HDTV	748
1.8.3.48	CbYCr422ToBGR	749
1.8.3.49	CbYCr422ToBGR 709HDTV	749
1.8.3.50	RGBToYCbCr420	751
1.8.3.51	YCbCr420ToRGB	751
1.8.3.52	YCbCr420ToRGBBatch	752
1.8.3.53	YCbCr420ToRGBBatchAdvanced	752
1.8.3.54	RGBToYCrCb420	753
1.8.3.55	YCrCb420ToRGB	754
1.8.3.56	BGRToYCbCr420	754
1.8.3.57	BGRToYCbCr420 709CSC	756
1.8.3.58	BGRToYCbCr420 709HDTV	757
1.8.3.59	BGRToYCrCb420 709CSC	757
1.8.3.60	YCbCr420ToBGR	758
1.8.3.61	YCbCr420ToBGRBatch	759
1.8.3.62	YCbCr420ToBGRBatchAdvanced	760
1.8.3.63	YCbCr420ToBGR 709CSC	761
1.8.3.64	YCbCr420ToBGR 709HDTV	761
1.8.3.65	BGRToYCrCb420	762
1.8.3.66	BGRToYCbCr411	763
1.8.3.67	BGRToYCbCr	764
1.8.3.68	YCbCr411ToBGR	765
1.8.3.69	YCbCr411ToRGB	766
1.8.3.70	RGBToXYZ	767
1.8.3.71	XYZToRGB	768
1.8.3.72	RGBToLUV	769
1.8.3.73	LUVToRGB	771
1.8.3.74	BGRToLab	773
1.8.3.75	LabToBGR	774
1.8.3.76	RGBToYCC	775
1.8.3.77	YCCToRGB	776
1.8.3.78	YCKToCMYK_JPEG	777
1.8.3.79	CMYKOrYCKJPEGToRGB	778
1.8.3.80	YCKJPEGOrCMYKToBGR	779
1.8.3.81	RGBToHLS	780
1.8.3.82	HLSToRGB	782
1.8.3.83	BGRToHLS	784

1.8.3.84	HLSToBGR	787
1.8.3.85	RBGToHSV	789
1.8.3.86	HSVToRGB	791
1.8.3.87	JPEG Color Conversion	792
1.9	Image Data Exchange And Initialization Functions	805
1.9.1	Set	805
1.9.1.1	Common parameters for nppiSet functions:	805
1.9.2	Masked Set	811
1.9.2.1	Common parameters for nppiSet_CXM functions:	811
1.9.3	Channel Set	814
1.9.3.1	Common parameters for nppiSet_CXC functions:	815
1.9.4	Copy	816
1.9.4.1	Common parameters for nppiCopy functions:	816
1.9.5	Masked Copy	821
1.9.5.1	Common parameters for nppiCopy_CXM functions:	822
1.9.6	Channel Copy	825
1.9.6.1	Common parameters for nppiCopy_CXC functions:	825
1.9.7	Extract Channel Copy	826
1.9.7.1	Common parameters for nppiCopy_CXC1 functions:	827
1.9.8	Insert Channel Copy	828
1.9.8.1	Common parameters for nppiCopy_C1CX functions:	829
1.9.9	Packed To Planar Channel Copy	830
1.9.9.1	Common parameters for nppiCopy_CXPX functions:	831
1.9.10	Planar To Packed Channel Copy	832
1.9.10.1	Common parameters for nppiCopy_PXCX functions:	833
1.9.11	Copy Constant Border	834
1.9.11.1	Common parameters for nppiCopyConstBorder functions:	834
1.9.12	Copy Replicate Border	839
1.9.12.1	Common parameters for nppiCopyReplicateBorder functions:	839
1.9.13	Copy Wrap Border	843
1.9.13.1	Common parameters for nppiCopyWrapBorder functions:	843
1.9.14	Copy Sub-Pixel	848
1.9.14.1	Common parameters for nppiCopySubPix functions:	848
1.9.15	Convert Bit Depth	851
1.9.16	Convert To Increased Bit Depth	852
1.9.16.1	Common parameters for nppiConvert to increased bit depth functions:	852
1.9.17	Convert To Decreased Bit Depth	860
1.9.17.1	Common parameters for nppiConvert to decreased bit depth functions:	860
1.9.18	Scale Bit Depth	869
1.9.19	Scale To Higher Bit Depth	869
1.9.19.1	Common parameters for nppiScale to higher bit depth functions:	869
1.9.20	Scale To Lower Bit Depth	872
1.9.20.1	Common parameters for nppiScale to lower bit depth functions:	872
1.9.21	Duplicate Channel	875
1.9.21.1	Common parameters for nppiDup functions:	875
1.9.22	Transpose	878
1.9.22.1	Common parameters for nppiTranspose functions:	878
1.9.23	Swap Channels	880
1.10	Image Filtering Functions	896
1.10.1	Image 1D Linear Filters	896
1.10.1.1	1DLinearFilter	896
1.10.1.2	Image Filter Column	896
1.10.1.3	Image Filter Column Border	900
1.10.1.4	Image Filter Column 32f	905

1.10.1.5	Image Filter Column Border 32f	907
1.10.1.6	Image Filter Row	911
1.10.1.7	Image Filter Row Border	914
1.10.1.8	Image Filter Row 32f	919
1.10.1.9	Image Filter Row Border 32f	921
1.10.1.10	Image Filter 1D Window Sum	925
1.10.1.11	Image Filter 1D Window Column Sum	925
1.10.1.12	Image Filter 1D Window Row Sum	928
1.10.1.13	Image Filter 1D Window Sum Border	931
1.10.1.14	Image Filter 1D Window Column Sum Border	931
1.10.1.15	Image Filter 1D Window Row Sum Border	935
1.10.2	Image Convolution	939
1.10.2.1	Convolution	939
1.10.2.2	Image Filter	940
1.10.2.3	Image Filter 32f	943
1.10.2.4	Image Filter Border	948
1.10.2.5	Image Filter Border 32f	952
1.10.3	2D Fixed Linear Filters	959
1.10.3.1	2D Fixed Linear Filters	959
1.10.3.2	Image Filter Box	960
1.10.3.3	Image Filter Box Border	963
1.10.3.4	Image Filter Box Border Advanced	966
1.10.3.5	Image Filter Threshold Adaptive Box Border	971
1.10.4	Rank Filters	972
1.10.4.1	Rank Filters	972
1.10.4.2	Image Filter Max	973
1.10.4.3	Image Filter Max Border	976
1.10.4.4	Image Filter Min	979
1.10.4.5	Image Filter Min Border	982
1.10.4.6	Image Filter Median	986
1.10.4.7	Image Filter Median Border	992
1.10.5	Fixed Filters	999
1.10.5.1	Fixed Filters	999
1.10.5.2	Image Filter Prewitt	1000
1.10.5.3	Image Filter Prewitt Border	1004
1.10.5.4	Image Filter Scharr	1010
1.10.5.5	Image Filter Scharr Border	1012
1.10.5.6	Image Filter Sobel	1014
1.10.5.7	Image Filter Sobel Border	1021
1.10.5.8	Image Filter Roberts	1032
1.10.5.9	Image Filter Roberts Border	1036
1.10.5.10	Image Filter Laplace	1042
1.10.5.11	Image Filter Laplace Border	1044
1.10.5.12	Image Filter Gauss	1048
1.10.5.13	Image Filter Gauss Advanced	1051
1.10.5.14	Image Filter Gauss Border	1055
1.10.5.15	Image Filter Advanced Gauss Border	1059
1.10.5.16	Image Filter Gauss Pyramid Layer Down Border	1064
1.10.5.17	Image Filter Gauss Pyramid Layer Up Border	1067
1.10.5.18	Image Filter Bilateral Gauss Border	1071
1.10.5.19	Image Filter High Pass	1075
1.10.5.20	Image Filter High Pass Border	1078
1.10.5.21	Image Filter Low Pass	1082
1.10.5.22	Image Filter Low Pass Border	1085

1.10.5.23	Image Filter Sharpen	.1089
1.10.5.24	Image Filter Sharpen Border	.1092
1.10.5.25	Image Filter Unsharp Border	.1096
1.10.5.26	Image Filter Wiener Border	.1103
1.10.5.27	Image Filter Gradient Vector Prewitt Border	.1107
1.10.5.28	Image Filter Gradient Vector Scharr Border	.1112
1.10.5.29	Image Filter Gradient Vector Sobel Border	.1118
1.10.6	Computer Vision Filtering Functions	.1123
1.10.6.1	Computer Vision	.1123
1.10.6.2	Image Filter Distance Transform	.1124
1.10.6.3	Image Filter Canny Border	.1167
1.10.6.4	Image Filter Harris Corners Border	.1169
1.10.6.5	Image Filter Hough Line	.1171
1.10.6.6	Image Filter Histogram Of Oriented Gradients Border	.1174
1.10.7	Image Filter Flood Fill	.1180
1.10.7.1	FloodFill	.1180
1.10.7.2	Flood Fill	.1180
1.10.7.3	Flood Fill Boundary	.1184
1.10.7.4	Flood Fill Range	.1188
1.10.7.5	Flood Fill Range Boundary	.1192
1.10.7.6	Flood Fill Gradient	.1197
1.10.7.7	Flood Fill Gradient Boundary	.1201
1.10.8	Label Markers	.1206
1.10.8.1	LabelMarkers	.1206
1.10.8.2	Label MarkersUF	.1207
1.10.8.3	Label MarkersUF Batch	.1209
1.10.8.4	Label MarkersUF Batch Advanced	.1211
1.10.8.5	Image Filter Compress Marker Labels	.1213
1.10.8.6	Image Filter Compressed Marker Labels Info	.1215
1.10.8.7	Image Filter Contour Pixel Interpolation	.1215
1.10.8.8	Contours Image Marching Squares Interpolation	.1215
1.10.9	Bound Segments	.1221
1.10.10	Watershed Segmentation	.1221
1.10.10.1	WatershedSegmentation	.1221
1.11	Image Geometry Transforms Functions	.1224
1.11.1	Geometric Transform API Specifics	.1224
1.11.2	Geometric Transforms and ROIs	.1224
1.11.3	Pixel Interpolation	.1224
1.11.4	Resize Error Codes	.1225
1.11.5	ResizeSqrPixel	.1225
1.11.6	Common parameters for npPiResizeSqrPackedPixel functions:	.1226
1.11.7	Common parameters for npPiResizeSqrPlanarPixel functions:	.1227
1.11.8	Resize	.1235
1.11.9	Common parameters for npPiResize packed pixel functions:	.1236
1.11.10	Common parameters for npPiResize planar pixel functions:	.1236
1.11.11	ResizeBatch	.1241
1.11.11.1	Common parameters for npPiResizeBatch functions:	.1243
1.11.11.2	Common parameters for npPiResizeBatchAdvanced functions:	.1243
1.11.12	Remap	.1248
1.11.13	Error Codes	.1249
1.11.13.1	Common parameters for npPiRemap packed pixel functions:	.1249
1.11.13.2	Common parameters for npPiRemap planar pixel functions:	.1249
1.11.14	Rotate	.1256
1.11.15	Rotate Error Codes	.1256

1.11.15.1	Common parameters for nppiRotate functions:	.1256
1.11.16	Rotate Utility Functions	.1259
1.11.17	Mirror	.1260
1.11.18	Mirror Error Codes	.1260
1.11.18.1	Common parameters for nppiMirror functions:	.1260
1.11.19	Affine Transforms	.1265
1.11.19.1	Affine Transforms	.1265
1.11.19.2	Affine Transform Utility Functions	.1266
1.11.19.3	Affine Transform	.1267
1.11.19.4	Affine Transform Batch	.1275
1.11.19.5	Common parameters for nppiWarpAffineBatch functions:	.1275
1.11.19.6	Backwards Affine Transform	.1278
1.11.19.7	Quad-Based Affine Transform	.1284
1.11.20	Perspective Transforms	.1291
1.11.20.1	Perspective Transform	.1291
1.11.20.2	Perspective Transform	.1293
1.11.20.3	Perspective Transform Batch	.1299
1.11.20.4	Backwards Perspective Transform	.1303
1.11.20.5	Quad-Based Perspective Transform	.1309
1.12	Image Linear Transforms Functions	.1316
1.12.1	Fourier Transforms	.1316
1.13	Image Morphological Operations	.1317
1.13.1	Dilation Functions	.1317
1.13.1.1	Image Dilate	.1317
1.13.1.2	Image Dilate Border	.1320
1.13.1.3	Image Dilate 3x3	.1323
1.13.1.4	Image Dilate 3x3 Border	.1326
1.13.2	Erosion Functions	.1328
1.13.2.1	Image Erode	.1328
1.13.2.2	Image Erode Border	.1331
1.13.2.3	Image Erode 3x3	.1334
1.13.2.4	Image Erode 3x3 Border	.1336
1.13.3	Image Complex Morphological Operations	.1339
1.13.3.1	Image Morph	.1339
1.13.3.2	Image Morph Get Buffer Size	.1339
1.13.3.3	Image Morph Close Border	.1341
1.13.3.4	Image Morph Open Border	.1344
1.13.3.5	Image Morph Top Hat Border	.1347
1.13.3.6	Image Morph Black Hat Border	.1350
1.13.3.7	Image Morph Gradient Border	.1353
1.14	Image Statistics Functions	.1356
1.14.1	CommonGetBufferHostSizeParameters	.1356
1.14.2	Image Sum	.1356
1.14.2.1	Sum	.1356
1.14.2.2	Common parameters for nppiSum functions include:	.1357
1.14.2.3	CommonSumGetBufferHostSizeParameters	.1360
1.14.3	Image Min	.1362
1.14.3.1	Min	.1362
1.14.3.2	Common parameters for nppiMin functions include:	.1362
1.14.3.3	CommonMinGetBufferHostSizeParameters	.1365
1.14.4	Image Min Index	.1367
1.14.4.1	MinIndx	.1367
1.14.4.2	Common parameters for nppiMinIndx functions include:	.1367
1.14.4.3	CommonMinIndxGetBufferHostSizeParameters	.1370

1.14.5	Image Max	.1372
1.14.5.1	Max	.1372
1.14.5.2	CommonMaxGetBufferHostSizeParameters	.1375
1.14.6	Image Max Index	.1377
1.14.6.1	MaxIndx	.1377
1.14.6.2	Common parameters for nppiMaxIndx functions include:	.1377
1.14.6.3	CommonMaxIndxGetBufferHostSizeParameters	.1380
1.14.7	Image MinMax	.1382
1.14.7.1	MinMax	.1382
1.14.7.2	Common parameters for nppiMinMax functions include:	.1382
1.14.7.3	CommonMinMaxGetBufferHostSizeParameters	.1385
1.14.8	Image Mean	.1387
1.14.8.1	Mean	.1387
1.14.8.2	Common parameters for nppiMean functions include:	.1387
1.14.8.3	CommonMeanGetBufferHostSizeParameters	.1391
1.14.9	Image Mean StdDev	.1394
1.14.9.1	Mean_StdDev	.1394
1.14.9.2	Common parameters for nppiMean_StdDev functions include:	.1394
1.14.9.3	MeanStdDevGetBufferHostSize	.1398
1.14.9.4	Common parameters for MeanStdDevGetBufferHostSize functions include:	.1398
1.14.10	Image Norms	.1400
1.14.10.1	Image Norms	.1400
1.14.10.2	Image Norm Inf	.1402
1.14.10.3	Image Norm L1	.1409
1.14.10.4	Image Norm L2	.1416
1.14.10.5	Image NormDiff Inf	.1422
1.14.10.6	Image NormDiff L1	.1430
1.14.10.7	Image NormDiff L2	.1437
1.14.10.8	Image NormRel Inf	.1444
1.14.10.9	Image NormRel L1	.1452
1.14.10.10	Image NormRel L2	.1459
1.14.11	Image DotProd	.1466
1.14.11.1	DotProd	.1466
1.14.11.2	Common parameters for nppiDotProd functions include:	.1466
1.14.11.3	CommonDotProdGetBufferHostSizeParameters	.1470
1.14.12	Image Count In Range	.1474
1.14.12.1	CountInRange	.1474
1.14.12.2	Common parameters for nppiCountInRange functions include:	.1475
1.14.12.3	CommonCountInRangeGetBufferHostSizeParameters	.1476
1.14.13	Image MaxEvery	.1478
1.14.13.1	MaxEvery	.1478
1.14.13.2	Common parameters for nppiMaxEvery functions include:	.1478
1.14.14	Image MinEvery	.1481
1.14.14.1	MinEvery	.1481
1.14.14.2	Common parameters for nppiMinEvery functions include:	.1481
1.14.15	Image Integral	.1484
1.14.15.1	Integral	.1484
1.14.16	Image Square Integral	.1485
1.14.16.1	SqrIntegral	.1485
1.14.17	Image RectStdDev	.1487
1.14.17.1	RectStdDev	.1487
1.14.18	Image Histogram Even	.1489
1.14.18.1	HistogramEven	.1489
1.14.18.2	Common parameters for nppiHistogramEven functions include:	.1489

1.14.18.3 Common parameters for nppiHistogramEvenGetBufferSize functions include:	.1492
1.14.19 Image Histogram Range	.1494
1.14.19.1 HistogramRange	.1494
1.14.19.2 Common parameters for nppiHistogramRange functions include:	.1494
1.14.20 Image Proximity	.1500
1.14.20.1 Image Proximity	.1500
1.14.21 Image Square Distance Full Norm	.1502
1.14.21.1 SqrDistanceFull_Norm	.1502
1.14.21.2 Common parameters for nppiSqrDistanceFull functions include:	.1502
1.14.22 Image Square Distance Same Norm	.1507
1.14.22.1 SqrDistanceSame_Norm	.1507
1.14.23 Image Square Distance Valid Norm	.1511
1.14.23.1 SqrDistanceValid_Norm	.1511
1.14.24 Image Cross Correlation Full Norm	.1515
1.14.24.1 CrossCorrFull_Norm	.1515
1.14.25 Image Cross Correlation Same Norm	.1520
1.14.25.1 CrossCorrSame_Norm	.1520
1.14.26 Image Cross Correlation Valid Norm	.1524
1.14.26.1 CrossCorrValid_Norm	.1524
1.14.27 Image Cross Correlation Valid	.1529
1.14.27.1 CrossCorrValid	.1529
1.14.28 Image Cross Correlation Full Norm Level	.1530
1.14.28.1 CrossCorrFull_NormLevel	.1530
1.14.28.2 CommonFullNormLevelGetBufferHostSizeParameters	.1535
1.14.29 Image Cross Correlation Same Norm Level	.1539
1.14.29.1 CrossCorrSame_NormLevel	.1539
1.14.29.2 CommonSameNormLevelGetBufferHostSizeParameters	.1544
1.14.30 Image Cross Correlation Valid Norm Level	.1548
1.14.30.1 CrossCorrValid_NormLevel	.1548
1.14.30.2 CommonValidNormLevelGetBufferHostSizeParameters	.1554
1.14.31 Image Cross Correlation Full Norm Level Advanced	.1557
1.14.31.1 CrossCorrFull_NormLevelAdvanced	.1557
1.14.32 Image Cross Correlation Same Norm Level Advanced	.1563
1.14.32.1 CrossCorrSame_NormLevelAdvanced	.1563
1.14.33 Image Cross Correlation Valid Norm Level Advanced	.1569
1.14.33.1 CrossCorrValid_NormLevelAdvanced	.1569
1.14.34 Image Quality Index	.1575
1.14.34.1 Image Quality Index	.1575
1.14.34.2 Common parameters for nppiQualityIndex functions include:	.1575
1.14.34.3 CommonQualityIndexGetBufferHostSizeParameters	.1577
1.14.35 Image Maximum Error	.1579
1.14.35.1 MaximumError	.1579
1.14.35.2 Common parameters for nppiMaximumError functions include:	.1579
1.14.35.3 CommonMaximumErrorGetBufferHostSizeParameters	.1587
1.14.36 Image Average Error	.1593
1.14.36.1 AverageError	.1593
1.14.36.2 CommonAverageErrorGetBufferHostSizeParameters	.1600
1.14.37 Image Maximum Relative Error	.1606
1.14.37.1 MaximumRelativeError	.1606
1.14.37.2 CommonMaximumRelativeErrorGetBufferHostSizeParameters	.1615
1.14.38 Image Average Relative Error	.1623
1.14.38.1 AverageRelativeError	.1623
1.14.38.2 CommonAverageRelativeErrorGetBufferHostSizeParameters	.1631
1.14.39 Image Quality Assessment IQA	.1639

1.14.39.1	IQA	.1639
1.14.40	Image Batch Quality Assessment	.1646
1.14.40.1	IQABatch	.1646
1.14.41	Image Advanced Batch Quality Assessment	.1654
1.14.41.1	IQABatchAdvanced	.1654
1.15	Image Threshold And Compare Operations	.1658
1.15.1	Image Threshold Operations	.1659
1.15.1.1	Threshold Operations	.1659
1.15.1.2	Image Threshold Greater Than Operations	.1664
1.15.1.3	Image Threshold Less Than Operations	.1670
1.15.1.4	Image Threshold Value Operations	.1675
1.15.1.5	Image Threshold Greater Than Value Operations	.1681
1.15.1.6	Image Threshold Less Than Value Operations	.1687
1.15.1.7	Image Fused AbsDiff Threshold Greater Than Value Operations	.1692
1.15.1.8	Image Threshold Less Than Value Greater Than Value Operations	.1694
1.15.2	Image Comparison Operations	.1701
1.15.2.1	Comparison Operations	.1701
1.15.2.2	Compare Images Operations	.1702
1.15.2.3	Compare Image With Constant Operations	.1705
1.15.2.4	Compare Image Differences With Epsilon Operations	.1709
1.15.2.5	Compare Image Difference To Constant With Epsilon Operations	.1710
1.16	Image Memory Management Functions	.1712
1.17	Signal Arithmetic And Logical Operations	.1720
1.17.1	Signal Arithmetic Functions	.1720
1.17.1.1	Arithmetic Operations	.1720
1.17.1.2	Signal AddC	.1720
1.17.1.3	Signal AddProductC	.1727
1.17.1.4	Signal MulC	.1727
1.17.1.5	Signal SubC	.1735
1.17.1.6	Signal SubCRev	.1741
1.17.1.7	Signal DivC	.1748
1.17.1.8	Signal DivCRev	.1753
1.17.1.9	Signal Add	.1754
1.17.1.10	Signal AddProduct	.1764
1.17.1.11	Signal Mul	.1766
1.17.1.12	Signal Sub	.1777
1.17.1.13	Signal Div	.1785
1.17.1.14	Signal Div Round	.1792
1.17.1.15	Signal Abs	.1794
1.17.1.16	Signal Square	.1797
1.17.1.17	Signal Square Root	.1801
1.17.1.18	Signal Cube Root	.1807
1.17.1.19	Signal Exp	.1808
1.17.1.20	Signal Ln	.1811
1.17.1.21	Signal 10Log10	.1814
1.17.1.22	Signal SumLn	.1815
1.17.1.23	Signal ArcTan	.1818
1.17.1.24	Signal Normalize	.1819
1.17.1.25	Signal Cauchy, CouchyD, And CouchyDD2	.1821
1.17.2	Logical And Shift Operations	.1822
1.17.2.1	Logical And Shift Operations	.1822
1.17.2.2	Signal AndC	.1823
1.17.2.3	Signal And	.1825
1.17.2.4	Signal OrC	.1827

1.17.2.5	Signal Or	.1829
1.17.2.6	Signal XorC	.1831
1.17.2.7	Signal Xor	.1833
1.17.2.8	Signal Not	.1835
1.17.2.9	Signal LShiftC	.1836
1.17.2.10	Signal RShiftC	.1840
1.18	Signal Conversion Functions	.1843
1.18.1	Signal Convert	.1843
1.18.1.1	Convert	.1843
1.18.2	Signal Threshold	.1852
1.18.2.1	Threshold	.1852
1.19	Signal Filtering Functions	.1872
1.19.1	Integral	.1872
1.20	Signal Initialization Functions	.1873
1.20.1	Signal Set	.1873
1.20.1.1	Set	.1873
1.20.2	Signal Zero	.1877
1.20.2.1	Zero	.1877
1.20.3	Signal Copy	.1880
1.20.3.1	Copy	.1880
1.21	Signal Statistical Functions	.1883
1.21.1	Signal Min Every Or Max Every	.1883
1.21.1.1	MinEvery And MaxEvery Functions	.1883
1.21.2	Signal Sum	.1886
1.21.2.1	Sum	.1886
1.21.3	Signal Maximum	.1892
1.21.3.1	Maximum	.1892
1.21.4	Signal Minimum	.1900
1.21.4.1	Minimum	.1900
1.21.5	Signal Mean	.1908
1.21.5.1	Mean	.1908
1.21.6	Signal StdDev	.1913
1.21.6.1	Standard Deviation	.1913
1.21.7	Signal Mean And StdDev	.1915
1.21.7.1	Mean And Standard Deviation	.1915
1.21.8	Signal MinMax	.1918
1.21.8.1	Minimum Maximum	.1918
1.21.9	Signal Norms	.1928
1.21.9.1	Signal Norm Inf	.1928
1.21.9.2	Signal Norm L1	.1932
1.21.9.3	Signal Norm L2	.1936
1.21.9.4	Signal Norm Inf NormDiff	.1941
1.21.9.5	Signal Norm L1 NormDiff	.1945
1.21.9.6	Signal Norm L2 NormDiff	.1950
1.21.10	Signal Dot Product	.1955
1.21.10.1	Dot Product	.1955
1.21.11	Signal Count In Range	.1971
1.21.11.1	Count In Range	.1971
1.21.12	Signal Count Zero Crossings	.1972
1.21.12.1	Count Zero Crossings	.1972
1.21.13	Signal Maximum Error	.1974
1.21.13.1	MaximumError	.1974
1.21.14	Signal Average Error	.1983
1.21.14.1	AverageError	.1983

1.21.15	Signal Maximum Relative Error	.1993
1.21.15.1	MaximumRelativeError	.1993
1.21.16	Signal Average Relative Error	2003
1.21.16.1	AverageRelativeError	2003
1.22	Signal Memory Management Functions	2013
1.22.1	Malloc	2013
1.22.2	Free	2016
Index		2017

Chapter 1. Indices and Search

- ▶ `genindex`
- ▶ `search`

1.1. What is NPP ?

NVIDIA NPP is a library of functions for performing CUDA accelerated 2D image and signal processing.

The primary set of functionality in the library focuses on image processing and is widely applicable for developers in these areas. NPP will evolve over time to encompass more of the compute heavy tasks in a variety of problem domains. The NPP library is written to maximize flexibility, while maintaining high performance.

NPP can be used in one of two ways:

- ▶ A stand-alone library for adding GPU acceleration to an application with minimal effort. Using this route allows developers to add GPU acceleration to their applications in a matter of hours.
- ▶ A cooperative library for interoperating with a developer's GPU code efficiently.

Either route allows developers to harness the massive compute resources of NVIDIA GPUs, while simultaneously reducing development times. After reading this Main Page it is recommended that you read the General API Conventions page below and either the Image-Processing Specific API Conventions page or Signal-Processing Specific API Conventions page depending on the kind of processing you expect to do. Finally, if you select the Modules tab at the top of this page you can find the kinds of functions available for the NPP operations that support your needs.

[What is NPP?](#) [general_conventions_lb](#) [nppi_conventions_lb](#) [npps_conventions_lb](#)

1.1.1. Files

NPP API is defined in the following files:

1.1.1.1 Header Files

- ▶ npp.h
- ▶ nppdefs.h
- ▶ nppcore.h
- ▶ nppi.h
- ▶ npps.h

All those header files are located in the following CUDA Toolkit's directory:

```
/include/
```

1.1.1.2 Library Files

NPP's functionality is split up into 3 distinct library groups:

- ▶ A core library (NPPC) containing basic functionality from the npp.h header file as well as common functionality used by the other two libraries.
- ▶ The image processing library NPPI. Any functions from the nppi.h header file or the various header files named "nppi_xxx.h" are bundled into the NPPI library.
- ▶ The signal processing library NPPS. Any function from the npps.h header file or the various header files named "npps_xxx.h" are bundled into the NPPS library.

On the Windows platform the NPP stub libraries are found in the CUDA Toolkit's library directory:

```
/lib/nppc.lib
```

```
/lib/nppial.lib
```

```
/lib/nppicc.lib
```

```
/lib/nppidei.lib
```

```
/lib/nppif.lib
```

```
/lib/nppig.lib
```

```
/lib/nppim.lib
```

```
/lib/nppist.lib
```

```
/lib/nppisu.lib
```

```
/lib/nppitc.lib
```

```
/lib/npps.lib
```

The matching DLLs are located in the CUDA Toolkit's binary directory. Example

```
* /bin/nppial64_111_<build_no>.dll // Dynamic image-processing library for 64-bit
↳ Windows.
```

On Linux platforms the dynamic libraries are located in the lib directory and the names include major and minor version numbers along with build numbers

```
* /lib/libnppc.so.11.1.<build_no> // NPP dynamic core library for Linux
```

1.1.2. Library Organization

Note: To minimize library loading and CUDA runtime startup times it is recommended to use the static library(s) whenever possible. To improve loading and runtime performance when using dynamic libraries, NPP provides a full set of NPPI sub-libraries. Linking to only the sub-libraries that contain functions that your application uses can significantly improve load time and runtime startup performance. Some NPPI functions make calls to other NPPI and/or NPPS functions internally so you may need to link to a few extra libraries depending on what function calls your application makes. The NPPI sub-libraries are split into sections corresponding to the way that NPPI header files are split. This list of sub-libraries is as follows:

- ▶ NPPC, NPP core library which MUST be included when linking any application, functions are listed in nppCore.h,
- ▶ NPPIAL, arithmetic and logical operation functions in npai_arithmetic_and_logical_operations.h,
- ▶ NPPICC, color conversion and sampling functions in npai_color_conversion.h,
- ▶ NPPIDEI, data exchange and initialization functions in npai_data_exchange_and_initialization.h,
- ▶ NPPIF, filtering and computer vision functions in npai_filtering_functions.h,
- ▶ NPPIG, geometry transformation functions found in npai_geometry_transforms.h,
- ▶ NPPIM, morphological operation functions found in npai_morphological_operations.h,
- ▶ NPPIST, statistics and linear transform in npai_statistics_functions.h and npai_linear_transforms.h,
- ▶ NPPISU, memory support functions in npai_support_functions.h,
- ▶ NPPITC, threshold and compare operation functions in npai_threshold_and_compare_operations.h,

For example, on Linux, to compile a small color conversion application foo using NPP against the dynamic library, the following command can be used:

```
nvcc foo.c -lnppc -lnppicc -o foo
```

Whereas to compile against the static NPP library, the following command has to be used:

```
nvcc foo.c -lnppc_static -lnppicc_static -o foo
```

It is also possible to use the native host C++ compiler. Depending on the host operating system, some additional libraries like pthread or dl might be needed on the linking line. The following command on Linux is suggested:

```
g++ foo.c -lnppc_static -lnppicc_static -lcudart_static -lpthread -ldl
-I <cuda-toolkit-path>/include -L <cuda-toolkit-path>/lib64 -o foo
```

NPP_Plus and NPP are stateless APIs. The default stream ID on any GPU device is 0. NPP_Plus and NPP applications using version 12.5 or greater MUST use the fully stateless application managed stream context interface described below. Also, the only function that remains in nppcore.h is nppGetVersion. The application can create one application managed stream context which can be modified on the fly or many application managed stream contexts each of which is unique to particular streams and/or GPU devices. The lifetime of application managed stream contexts are determined by the application. In general it is recommended to limit the number of unique application managed stream contexts to the number of GPUs in use as reinitializing an application managed stream context can have high overhead due to the need for CUDA device calls. However the stream ID of a single application managed stream context can be changed on the fly as needed with no extra overhead. It is the responsibility of the application to detect and respond to GPU resets if they happen and to reinitialize any existing application managed stream contexts. Calling older NPP non application managed stream context functions will now return the new NPP_STREAM_CONTEXT_ERROR value.

All NPP functions should be thread safe.

Note: In NPP 12.4 in order to support very large images NPP 12.4 and beyond has change the return value from GetBufferSize calls for statistics functions and a few others from int to size_t. Signal lengths have also changed from int to size_t. This should be the only change required for applications to be rebuilt using these versions of NPP.

Note: NPP 12.1 is the last release of NPP which will support NPP API calls that do not contain NPP stream context parameters. Also NPP will soon release an API variant that provides collapsed combined parameter versions of many API calls. For example a call like nppiAdd_8u_C3R_Ctx(pSrc1, nSrc1Step, pSrc2, nSrc2Step, pDst, nDstStep, oSizeROI, nppStreamCtx) will become nppiAdd_Ctx(NPP_8U, NPP_CH_3, pSrc1, nSrc1Step, pSrc2, nSrcStep2, pDst, nDstStep, oSizeROI, nppStreamCtx). This makes adding support for new data types and number of channels simpler as well as significantly reducing redundant documentation.

Note: New to NPP 11.6 are

```
nppiContoursImageMarchingSquaresInterpolation_32f_C1R_Ctx
```

```
nppiContoursImageMarchingSquaresInterpolation_64f_C1R_Ctx
```

Note: New to NPP 10.1 is support for the fp16 (__half) data type in GPU architectures of Volta and beyond in some NPP image processing functions. NPP image functions that support pixels of __half data types have function names of type 16f and pointers to pixels of that data type need to be passed to NPP as NPP data type *Npp16f*. Here is an example of how to pass image pointers of type __half to an NPP 16f function that should work on all compilers including Armv7.

```
nppiAdd_16f_C3R_Ctx(reinterpret_cast<const Npp16f *>((const void *) (pSrc1Data)),
↪ nSrc1Pitch,
                    reinterpret_cast<const Npp16f *>((const void *) (pSrc2Data)),
↪ nSrc2Pitch,
                    reinterpret_cast<Npp16f *>((void *) (pDstData)), nDstPitch,
                    oDstROI, nppStreamCtx);
```

Application Managed Stream Context Application Managed Stream Context

Note: Also new to NPP 10.1 is support for application managed stream contexts. Application managed stream contexts make NPP truly stateless internally allowing for rapid, no overhead, stream context switching. While it is recommended that all new NPP application code use application managed stream contexts, existing application code can continue to use nppSetStream() and nppGetStream() to manage stream contexts (also with no overhead now) but over time NPP will likely deprecate the older

non-application managed stream context API. Both the new and old stream management techniques can be intermixed in applications but any NPP calls using the old API will use the stream set by the most recent call to `nppSetStream()` and `nppGetStream()` calls will also return that stream ID. All NPP function names ending in `_Ctx` expect application managed stream contexts to be passed as a parameter to that function. The new *NppStreamContext* application managed stream context structure is defined in `nppdefs.h` and should be initialized by the application to the Cuda device ID and values associated with a particular stream. Applications can use multiple fixed stream contexts or change the values in a particular stream context on the fly whenever a different stream is to be used.

Note: NPP 10.2 and beyond contain an additional element in the *NppStreamContext* structure named `nStreamFlags` which MUST also be initialized by the application. Failure to do so could unnecessarily reduce NPP performance in some functions.

Note: NPP does not support non blocking streams on Windows for devices working in WDDM mode.

Note that some of the “GetBufferSize” style functions now have application managed stream contexts associated with them and should be used with the same stream context that the associated application managed stream context NPP function will use.

Note that NPP does minimal checking of the parameters in an application managed stream context structure so it is up to the application to assure that they are correct and valid when passed to NPP functions.

Note that NPP has deprecated the `nppicom` JPEG compression library as of NPP 11.0, use the `NVJPEG` library instead.

1.1.3. Supported NVIDIA Hardware

NPP runs on all CUDA capable NVIDIA hardware. For details please see http://www.nvidia.com/object/cuda_learn_products.html

1.2. General Conventions

1.2.1. Memory Management

The design of all the NPP functions follows the same guidelines as other NVIDIA CUDA libraries like `cuFFT` and `cuBLAS`. That is that all pointer arguments in those APIs are device pointers.

This convention enables the individual developer to make smart choices about memory management that minimize the number of memory transfers. It also allows the user the maximum flexibility regarding which of the various memory transfer mechanisms offered by the CUDA runtime is used, e.g. synchronous or asynchronous memory transfers, zero-copy and pinned memory, etc.

The most basic steps involved in using NPP for processing data is as follows:

1. Transfer input data from the host to device using

```
cudaMemcpy( . . . )
```

2. Process data using one or several NPP functions or custom CUDA kernels
3. Transfer the result data from the device to the host using

```
cudaMemCpy(...)
```

1.2.1.1 Scratch Buffer and Host Pointer

Some primitives of NPP require additional device memory buffers (scratch buffers) for calculations, e.g. signal and image reductions (Sum, Max, Min, MinMax, etc.). In order to give the NPP user maximum control regarding memory allocations and performance, it is the user's responsibility to allocate and delete those temporary buffers. For one this has the benefit that the library will not allocate memory unbeknownst to the user. It also allows developers who invoke the same primitive repeatedly to allocate the scratch only once, improving performance and potential device-memory fragmentation.

Scratch-buffer memory is unstructured and may be passed to the primitive in uninitialized form. This allows for reuse of the same scratch buffers with any primitive require scratch memory, as long as it is sufficiently sized.

The minimum scratch-buffer size for a given primitive (e.g. `nppsSum_32f_Ctx()`) can be obtained by a companion function (e.g. `nppsSumGetBufferSize_32f_Ctx()`). The buffer size is returned via a host pointer as allocation of the scratch-buffer is performed via CUDA runtime host code.

An example to invoke signal sum primitive and allocate and free the necessary scratch memory:

```
// pSrc, pSum, pDeviceBuffer are all device pointers.
NppStreamContext nppStreamCtx; // previously initialized
Npp32f * pSrc;
Npp32f * pSum;
Npp8u * pDeviceBuffer;
size_t nLength = 1024;

// Allocate the device memroy.
cudaMalloc((void **)&pSrc, sizeof(Npp32f) * nLength);
nppsSet_32f_Ctx(1.0f, pSrc, nLength, nppStreamCtx);
cudaMalloc((void **)&pSum, sizeof(Npp32f) * 1);

// Compute the appropriate size of the scratch-memory buffer, note that nBufferSize
// and nLength data types have changed from int to size_t.
size_t nBufferSize;
nppsSumGetBufferSize_32f_Ctx(nLength, &nBufferSize, nppStreamCtx);
// Allocate the scratch buffer
cudaMalloc((void **)&pDeviceBuffer, nBufferSize);

// Call the primitive with the scratch buffer
nppsSum_32f_Ctx(pSrc, nLength, pSum, pDeviceBuffer, nppStreamCtx);
Npp32f nSumHost;
cudaMemcpy(&nSumHost, pSum, sizeof(Npp32f) * 1, cudaMemcpyDeviceToHost);
printf("sum = %f\n", nSumHost); // nSumHost = 1024.0f;

// Free the device memory
cudaFree(pSrc);
cudaFree(pDeviceBuffer);
cudaFree(pSum);
```

1.2.2. Function Naming

Since NPP is a C API and therefore does not allow for function overloading for different data-types the NPP naming convention addresses the need to differentiate between different flavors of the same algorithm or primitive function but for various data types. This disambiguation of different flavors of a primitive is done via a suffix containing data type and other disambiguating information.

In addition to the flavor suffix, all NPP functions are prefixed with by the letters “npp”. Primitives belonging to NPP’s image-processing module add the letter “i” to the npp prefix, i.e. are prefixed by “nppi”. Similarly signal-processing primitives are prefixed with “npps”.

The general naming scheme is:

```
npp<module info><PrimitiveName>_<data-type info>[_<additional flavor info>](
↪<parameter list>)
```

The data-type information uses the same names as the Basic NPP Data Types. For example the data-type information “8u” would imply that the primitive operates on *Npp8u* data.

If a primitive consumes different type data from what it produces, both types will be listed in the order of consumed to produced data type.

Details about the “additional flavor information” is provided for each of the NPP modules, since each problem domain uses different flavor information suffixes.

1.2.3. Integer Result Scaling

NPP signal processing and imaging primitives often operate on integer data. This integer data is usually a fixed point fractional representation of some physical magnitue (e.g. luminance). Because of this fixed-point nature of the representation many numerical operations (e.g. addition or multiplication) tend to produce results exceeding the original fixed-point range if treated as regular integers.

In cases where the results exceed the original range, these functions clamp the result values back to the valid range. E.g. the maximum positive value for a 16-bit unsigned integer is 32767. A multiplication operation of $4 * 10000 = 40000$ would exceed this range. The result would be clamped to be 32767.

To avoid the level of lost information due to clamping most integer primitives allow for result scaling. Primitives with result scaling have the “Sfs” suffix in their name and provide a parameter “nScaleFactor” that controls the amount of scaling. Before the results of an operation are clamped to the valid output-data range by multiplying them with $2^{-nScaleFactor}$.

Example: The primitive *nppsSqr_8u_Sfs_Ctx()* computes the square of 8-bit unsigned sample values in a signal (1D array of values). The maximum value of a 8-bit value is 255. The square of $255^2 = 65025$ which would be clamped to 255 if no result scaling is performed. In order to map the maximum value of 255 to 255 in the result, one would specify an integer result scaling factor of 8, i.e. multiply each result with $2^{-8} = \frac{1}{2^8} = \frac{1}{256}$. The final result for a signal value of 255 being squared and scaled would be:

$$255^2 \cdot 2^{-8} = 254.00390625$$

which would be rounded to a final result of 254.

A medium gray value of 128 would result in

$$128^2 * 2^{-8} = 64$$

1.2.4. Rounding Modes

Many NPP functions require converting floating-point values to integers. The *Rounding Modes* enum lists NPP's supported rounding modes. Not all primitives in NPP that perform rounding as part of their functionality allow the user to specify the round-mode used. Instead they use NPP's default rounding mode, which is NPP_RND_FINANCIAL.

1.2.4.1 Rounding Mode Parameter

A subset of NPP functions performing rounding as part of their functionality do allow the user to specify which rounding mode is used through a parameter of the *Rounding Modes* type.

1.3. Image Processing Conventions

1.3.1. Function Naming

Image processing related functions use a number of suffixes to indicate various different flavors of a primitive beyond just different data types. The flavor suffix uses the following abbreviations:

- ▶ "A" if the image is a 4 channel image this indicates the result alpha channel is not affected by the primitive.
- ▶ "Cn" the image consists of n channel packed pixels, where n can be 1, 2, 3 or 4.
- ▶ "Pn" the image consists of n separate image planes, where n can be 1, 2, 3 or 4.
- ▶ "C" (following the channel information) indicates that the primitive only operates on one of the color channels, the "channel-of-interest". All other output channels are not affected by the primitive.
- ▶ "I" indicates that the primitive works "in-place". In this case the image-data pointer is usually named pSrcDst to indicate that the image data serves as source and destination at the same time.
- ▶ "M" indicates "masked operation". These types of primitives have an additional "mask image" as input. Each pixel in the destination image corresponds to a pixel in the mask image. Only pixels with a corresponding non-zero mask pixel are being processed.
- ▶ "R" indicates the primitive operates only on a rectangular region_of_interest or "ROI". All ROI primitives take an additional input parameter of type *NppiSize*, which specifies the width and height of the rectangular region that the primitive should process. For details on how primitives operate on ROIs see: :ref: 'roi_specification'.
- ▶ "Sfs" indicates the result values are processed by fixed scaling and saturation before they're written out.

The suffixes above always appear in alphabetical order. E.g. a 4 channel primitive not affecting the alpha channel with masked operation, in place and with scaling/saturation and ROI would have the postfix: “AC4IMRSfs”.

1.3.2. Image Data

Image data is passed to and from NPPI primitives via a pair of parameters:

1. A pointer to the image’s underlying data type.
2. A line step in bytes (also sometimes called line stride).

The general idea behind this fairly low-level way of passing image data is ease-of-adoption into existing software projects:

- ▶ Passing a raw pointer to the underlying pixel data type, rather than structured (by color) channel pixel data allows usage of the function in a wide variety of situations avoiding risky type cast or expensive image data copies.
- ▶ Passing the data pointer and line step individually rather than a higher-level image struct again allows for easy adoption by not requiring a specific image representation and thus avoiding awkward packing and unpacking of image data from the host application to an NPP specific image representation.

1.3.2.1 Line Step

The line step (also called “line stride” or “row step”) allows lines of oddly sized images to start on well-aligned addresses by adding a number of unused bytes at the ends of the lines. This type of line padding has been common practice in digital image processing for a long time and is not particular to GPU image processing.

The line step is the number of bytes in a line **including the padding**. An other way to interpret this number is to say that it is the number of bytes between the first pixels of successive rows in the image, or generally the number of bytes between two neighboring pixels in any column of pixels.

The general reason for the existence of the line step it is that uniformly aligned rows of pixel enable optimizations of memory-access patterns.

Even though all functions in NPP will work with arbitrarily aligned images, best performance can only be achieved with well aligned image data. Any image data allocated with the NPP image allocators or the 2D memory allocators in the CUDA runtime, is well aligned.

Particularly on older CUDA capable GPUs it is likely that the performance decrease for misaligned data is substantial (orders of magnitude).

All image data passed to NPPI primitives requires a line step to be provided. It is important to keep in mind that this line step is always specified in terms of bytes, not pixels.

1.3.2.2 Parameter Names for Image Data

There are three general cases of image-data passing throughout NPP detailed in the following sections.

1.3.2.2.1 Passing Source-Image Data

Those are images consumed by the algorithm.

1.3.2.2.2 Source-Image Pointer

The source image data is generally passed via a pointer named

`pSrc`

The source image pointer is generally defined constant, enforcing that the primitive does not change any image data pointed to by that pointer. E.g.

```
nppiPrimitive_32s_C1R_Ctx(const Npp32s * pSrc, ...)
```

In case the primitive consumes multiple images as inputs the source pointers are numbered like this:

`pSrc1, pSrc2, ...`

1.3.2.2.3 Source-Batch-Images Pointer

The batch of source images data is generally passed via a pointer of *NppiImageDescriptor* type named

`pSrcBatchList`

The source batch pointer is generally defined constant, enforcing that the primitive does not change any source data pointed to by that pointer. E.g.

```
nppiYUVToRGBBatch_8u_C3R_Ctx(NppiSize oSizeROI, const NppiImageDescriptor*  
↪ pSrcBatchList, ...)
```

All primitives processing batch data require providing the size of the batch in a separate parameter.

1.3.2.2.4 Source-Planar-Image Pointer Array

The planar source image data is generally passed via an array of pointers named

`pSrc[]`

The planar source image pointer array is generally defined a constant array of constant pointers, enforcing that the primitive does not change any image data pointed to by those pointers. E.g.

```
nppiPrimitive_8u_P3R_Ctx(const Npp8u * const pSrc[3], ...)
```

Each pointer in the array points to a different image plane.

1.3.2.2.5 Source-Planar-Image Pointer

The multiple plane source image data is passed via a set of pointers named

```
pSrc1, pSrc2, ...
```

The planar source image pointer is generally defined as one of a set of constant pointers with each pointer pointing to a different input image plane.

1.3.2.2.6 Source-Image Line Step

The source image line step is the number of bytes between successive rows in the image. The source image line step parameter is

```
nSrcStep
```

or in the case of multiple source images

```
nSrcStep1, nSrcStep2, ...
```

1.3.2.2.7 Source-Planar-Image Line Step Array

The source planar image line step array is an array where each element of the array contains the number of bytes between successive rows for a particular plane in the input image. The source planar image line step array parameter is

```
rSrcStep[ ]
```

1.3.2.2.8 Source-Planar-Image Line Step

The source planar image line step is the number of bytes between successive rows in a particular plane of the multiplane input image. The source planar image line step parameter is

```
nSrcStep1, nSrcStep2, ...
```

1.3.2.2.9 Passing Destination-Image Data

Those are images produced by the algorithm.

1.3.2.2.10 Destination-Image Pointer

The destination image data is generally passed via a pointer named

```
pDst
```

In case the primitive generates multiple images as outputs the destination pointers are numbered like this:

```
pDst1, pDst2, ...
```

1.3.2.2.11 Destination-Batch-Images Pointer

The batch of destination images data is generally passed via a pointer of *NppImageDescriptor* type named

```
pDstBatchList
```

All primitives processing batch data require providing the size of the batch in a separate parameter.

1.3.2.2.12 Destination-Planar-Image Pointer Array

The planar destination image data pointers are generally passed via an array of pointers named

```
pDst[ ]
```

Each pointer in the array points to a different image plane.

1.3.2.2.13 Destination-Planar-Image Pointer

The destination planar image data is generally passed via a pointer to each plane of a multiplane output image named

```
pDst1, pDst2, ...
```

1.3.2.2.14 Destination-Image Line Step

The destination image line step parameter is

```
nDstStep
```

or in the case of multiple destination images

```
nDstStep1, nDstStep2, ...
```

1.3.2.2.15 Destination-Planar-Image Line Step

The destination planar image line step is the number of bytes between successive rows for a particular plane in a multiplane output image. The destination planar image line step parameter is

```
nDstStep1, nDstStep2, ...
```

1.3.2.2.16 Passing In-Place Image Data

1.3.2.2.17 In-Place Image Pointer

In the case of in-place processing, source and destination are served by the same pointer and thus pointers to in-place image data are called:

```
pSrcDst
```

1.3.2.2.18 In-Place-Image Line Step

The in-place line step parameter is

```
nSrcDstStep
```

1.3.2.2.19 Passing Mask-Image Data

Some image processing primitives have variants supporting `masked_operation`.

1.3.2.2.20 Mask-Image Pointer

The mask-image data is generally passed via a pointer named

```
pMask
```

1.3.2.2.21 Mask-Image Line Step

The mask-image line step parameter is

```
nMaskStep
```

1.3.2.2.22 Passing Channel-of-Interest Data

Some image processing primitives support `channel_of_interest`.

1.3.2.2.23 Channel_of_Interest Number

The channel-of-interest data is generally an integer (either 1, 2, or 3):

```
nCOI
```

1.3.2.3 Image Data Alignment Requirements

NPP requires pixel data to adhere to certain alignment constraints.

For 2 and 4 channel images the following alignment requirement holds:

```
data_pointer % (\#channels * sizeof(channel type)) == 0
```

E.g. a 4 channel image with underlying type `Npp8u` (8-bit unsigned) would require all pixels to fall on addresses that are multiples of 4 (4 channels * 1 byte size).

As a logical consequence of all pixels being aligned to their natural size the image line steps of 2 and 4 channel images also need to be multiples of the pixel size.

For 1 and 3 channel images only require that pixel pointers are aligned to the underlying data type, i.e. `pData % sizeof(data type) == 0`. And consequentially line steps are also held to this requirement.

1.3.2.4 Image Data Related Error Codes

All NPPI primitives operating on image data validate the image-data pointer for proper alignment and test that the point is not null. They also validate the line stride for proper alignment and guard against the step being less or equal to 0. Failed validation results in one of the following error codes being returned and the primitive not being executed: `NPP_STEP_ERROR` is returned if the data step is 0 or negative. `NPP_NOT_EVEN_STEP_ERROR` is returned if the line step is not a multiple of the pixel size for 2 and 4 channel images. `NPP_NULL_POINTER_ERROR` is returned if the image-data pointer is 0 (NULL). `NPP_ALIGNMENT_ERROR` if the image-data pointer address is not a multiple of the pixel size for 2 and 4 channel images.

1.3.3. Region-Of-Interest (ROI)

In practice processing a rectangular sub-region of an image is often more common than processing complete images. The vast majority of NPP's image-processing primitives allow for processing of such sub regions also referred to as regions-of-interest or ROIs.

All primitives supporting ROI processing are marked by a "R" in their name suffix. In most cases the ROI is passed as a single `NppiSize` struct, which provides the width and height of the ROI. This raises the question how the primitive knows where in the image this rectangle of (width, height) is located. The "start pixel" of the ROI is implicitly given by the image-data pointer. I.e. instead of explicitly passing a pixel coordinate for the upper-left corner (lowest memory address), the user simply offsets the image-data pointers to point to the first pixel of the ROI.

In practice this means that for an image (pSrc, nSrcStep) and the start-pixel of the ROI being at location (x, y), one would pass

```
pSrcOffset = pSrc + y * nSrcStep + x * PixelSize;
```

as the image-data source to the primitive. PixelSize is typically computed as

```
PixelSize = NumberOfColorChannels * sizeof(PixelDataType).
```

E.g. for a primitive like *nppiSet_16s_C4R_Ctx()* we would have

- ▶ NumberOfColorChannels == 4;
- ▶ sizeof(Npp16s) == 2;
- ▶ and thus PixelSize = 4 * 2 = 8;

1.3.3.1 ROI Related Error Codes

All NPPI primitives operating on ROIs of image data validate the ROI size and image's step size. Failed validation results in one of the following error codes being returned and the primitive not being executed: NPP_SIZE_ERROR is returned if either the ROI width or ROI height are negative. NPP_STEP_ERROR is returned if the ROI width exceeds the image's line step. In mathematical terms $(\text{widthROI} * \text{PixelSize}) > \text{nLinStep}$ indicates an error.

1.3.4. Masked Operation

Some primitive support masked operation. An "M" in the suffix of those variants indicates masked operation. Primitives supporting masked operation consume an additional input image provided via a mask_image_pointer and mask_image_line_step. The mask image is interpreted by these primitives as a boolean image. The values of type Npp8u are interpreted as boolean values where a values of 0 indicates false, any non-zero values true.

Unless otherwise indicated the operation is only performed on pixels where its spatially corresponding mask pixel is true (non-zero). E.g. a masked copy operation would only copy those pixels in the ROI that have corresponding non-zero mask pixels.

1.3.5. Channel-of-Interest API

Some primitives allow restricting operations to a single channel of interest within a multi-channel image. These primitives are suffixed with the letter "C" (after the channel information, e.g. *nppiCopy_8u_C3CR_Ctx()*). The channel-of-interest is generally selected by offsetting the image-data pointer to point directly to the channel- of-interest rather than the base of the first pixel in the ROI. Some primitives also explicitly specify the selected channel number and pass it via an integer, e.g. *nppiMean_StdDev_8u_C3CR_Ctx()*.

1.3.5.1 Select-Channel Source-Image Pointer

This is a pointer to the channel-of-interest within the first pixel of the source image. E.g. if `pSrc` is the pointer to the first pixel inside the ROI of a three channel image. Using the appropriate select-channel copy primitive one could copy the second channel of this source image into the first channel of a destination image given by `pDst` by offsetting the pointer by one:

```
nppiCopy_8u_C3CR_Ctx(pSrc + 1, nSrcStep, pDst, nDstStep, oSizeROI);
```

1.3.5.2 Select-Channel Source-Image

Some primitives allow the user to select the channel-of-interest by specifying the channel number (`nCOI`). This approach is typically used in the image statistical functions. For example,

```
nppiMean_StdDev_8u_C3CR_Ctx(pSrc, nSrcStep, oSizeROI, nCOI, pDeviceBuffer, pMean,  
↪ pStdDev );
```

The channel-of-interest number can be either 1, 2, or 3.

1.3.5.3 Select-Channel Destination-Image Pointer

This is a pointer to the channel-of-interest within the first pixel of the destination image. E.g. if `pDst` is the pointer to the first pixel inside the ROI of a three channel image. Using the appropriate select-channel copy primitive one could copy data into the second channel of this destination image from the first channel of a source image given by `pSrc` by offsetting the destination pointer by one:

```
nppiCopy_8u_C3CR_Ctx(pSrc, nSrcStep, pDst + 1, nDstStep, oSizeROI);
```

1.3.6. Source-Image Sampling

A large number of NPP image-processing functions consume at least one source image and produce an output image (e.g. `nppiAddC_8u_C1RSfs_Ctx()` or `nppiFilterBox_8u_C1R_Ctx()`). All NPP functions falling into this category also operate on ROIs (see :ref: _roi_specification) which for these functions should be considered to describe the destination ROI. In other words the ROI describes a rectangular region in the destination image and all pixels inside of this region are being written by the function in question.

In order to use such functions successfully it is important to understand how the user defined destination ROI affects which pixels in the input image(s) are being read by the algorithms. To simplify the discussion of ROI propagation (i.e. given a destination ROI, what are the ROIs in the source(s)), it makes sense to distinguish two major cases:

1. Point-Wise Operations: These are primitives like `nppiAddC_8u_C1RSfs_Ctx()`. Each output pixel requires exactly one input pixel to be read.
2. Neighborhood Operations: These are primitives like `nppiFilterBox_8u_C1R_Ctx()`, which require a group of pixels from the source image(s) to be read in order to produce a single output.

1.3.6.1 Point-Wise Operations

As mentioned above, point-wise operations consume a single pixel from the input image (or a single pixel from each input image, if the operation in question has more than one input image) in order to produce a single output pixel.

1.3.6.2 Neighborhood Operations

In the case of neighborhood operations a number of input pixels (a “neighborhood” of pixels) is read in the input image (or images) in order to compute a single output pixel. All of the functions for `image_filtering_functions` and `image_morphological_operations` are neighborhood operations.

Most of these functions have parameters that affect the size and relative location of the neighborhood: a mask-size structure and an anchor-point structure. Both parameters are described in more detail in the next subsections.

1.3.6.2.1 Mask-Size Parameter

Many NPP neighborhood operations allow the user to specify the size of the neighborhood via a parameter usually named `oMaskSize` of type `NppiSize`. In those cases the neighborhood of pixels read from the source(s) is exactly the size of the mask. Assuming the mask is anchored at location (0, 0) (see `anchor_point_parameter` below) and has a size of (w, h), i.e.

```
assert(oMaskSize.w == w);
assert(oMaskSize.h == h);
assert(oAnchor.x == 0);
assert(oAnchor.y == 0);
```

a neighborhood operation would read the following source pixels in order to compute destination pixel $D_{i,j}$:

$$\begin{array}{cccc}
 S_{i,j} & S_{i,j+1} & \cdots & S_{i,j+w-1} \\
 S_{i+1,j} & S_{i+1,j+1} & \cdots & S_{i+1,j+w-1} \\
 \vdots & \vdots & \ddots & \vdots \\
 S_{i+h-1,j} & S_{i+h-1,j+1} & \cdots & S_{i+h-1,j+w-1}
 \end{array}$$

1.3.6.2.2 Anchor-Point Parameter

Many NPP primitives performing neighborhood operations allow the user to specify the relative location of the neighborhood via a parameter usually named `oAnchor` of type `NppiPoint`. Using the anchor a developer can choose the position of the mask (see `mask_size_parameter`) relative to current pixel index.

Using the same example as in `mask_size_parameter`, but this time with an anchor position of (a, b):

```
assert(oMaskSize.w == w);
assert(oMaskSize.h == h);
assert(oAnchor.x == a);
assert(oAnchor.y == b);
```

the following pixels from the source image would be read:

$$\begin{array}{cccc} S_{i-a,j-b} & S_{i-a,j-b+1} & \cdots & S_{i-a,j-b+w-1} \\ S_{i-a+1,j-b} & S_{i-a+1,j-b+1} & \cdots & S_{i-a+1,j-b+w-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i-a+h-1,j-b} & S_{i-a+h-1,j-b+1} & \cdots & S_{i-a+h-1,j-b+w-1} \end{array}$$

1.3.6.2.3 Sampling Beyond Image Boundaries

NPP primitives in general and NPP neighborhood operations in particular require that all pixel locations read and written are valid and within the boundaries of the respective images. Sampling outside of the defined image data regions results in undefined behavior and may lead to system instability.

This poses a problem in practice: when processing full-size images one cannot choose the destination ROI to be the same size as the source image. Because neighborhood operations read pixels from an enlarged source ROI, the destination ROI must be shrunk so that the expanded source ROI does not exceed the source image's size OR if the neighborhood operation function supports a Border version then this version can be used without ROI adjustment with the appropriate border protection mode selected.

For cases where this “shrinking” of the destination image size is unacceptable and a Border version of the function is not available, NPP provides a set of border-expanding Copy primitives. E.g. [*nppiCopyConstBorder_8u_C1R_Ctx\(\)*](#), [*nppiCopyReplicateBorder_8u_C1R_Ctx\(\)*](#) and [*nppiCopyWrapBorder_8u_C1R_Ctx\(\)*](#). The user can use these primitives to “expand” the source image's size using one of the three expansion modes. The expanded image can then be safely passed to a neighborhood operation producing a full-size result.

1.4. Signal Processing Conventions

1.4.1. Signal Data

Signal data is passed to and from NPPS primitives via a pointer to the signal's data type.

The general idea behind this fairly low-level way of passing signal data is ease-of-adoption into existing software projects:

- Passing the data pointer rather than a higher-level signal struct allows for easy adoption by not requiring a specific signal representation (that could include total signal size offset, or other additional information). This avoids awkward packing and unpacking of signal data from the host application to an NPP specific signal representation.

1.4.1.1 Parameter Names for Signal Data

There are three general cases of image-data passing throughout NPP detailed in the following sections.

Those are signals consumed by the algorithm.

1.4.1.1.1 Source Signal Pointer

The source signal data is generally passed via a pointer named

```
pSrc
```

The source signal pointer is generally defined constant, enforcing that the primitive does not change any image data pointed to by that pointer. E.g.

```
nppsPrimitive_32s(const Npp32s * pSrc, ...)
```

In case the primitive consumes multiple signals as inputs the source pointers are numbered like this:

```
pSrc1, pSrc2, ...
```

1.4.1.1.2 Destination Signal Pointer

The destination signal data is generally passed via a pointer named

```
pDst
```

In case the primitive consumes multiple signals as inputs the source pointers are numbered like this:

```
pDst1, pDst2, ...
```

1.4.1.1.3 In-Place Signal Pointer

In the case of in-place processing, source and destination are served by the same pointer and thus pointers to in-place signal data are called:

```
pSrcDst
```

1.4.1.2 Signal Data Alignment Requirements

NPP requires signal sample data to be naturally aligned, i.e. any pointer

```
NppType * p;
```

to a sample in a signal needs to fulfill:

```
assert(p % sizeof(p) == 0);
```

1.4.1.3 Signal Data Related Error Codes

All NPPI primitives operating on signal data validate the signal-data pointer for proper alignment and test that the point is not null.

Failed validation results in one of the following error codes being returned and the primitive not being executed: `NPP_NULL_POINTER_ERROR` is returned if the image-data pointer is 0 (NULL). `NPP_ALIGNMENT_ERROR` if the signal-data pointer address is not a multiple of the signal's data-type size.

1.4.2. Signal Length

The vast majority of NPPS functions take a

`nLength`

parameter that tells the primitive how many of the signal's samples starting from the given data pointer are to be processed.

1.4.2.1 Length Related Error Codes

All NPPS primitives taking a length parameter validate this input.

Failed validation results in the following error code being returned and the primitive not being executed: `NPP_SIZE_ERROR` is returned if the length is negative.

1.5. Data Types, Structs, Enums, and Constants

struct **Npp16f**

Workarounds for `cuda_fp16.h` C incompatibility *Npp16f*.

Public Members

short **fp16**

Original Cuda fp16 data size and format.

struct **Npp16f_2**

Npp16f_2.

Public Members

short **fp16_0**

Original Cuda fp16 data size and format.

short **fp16_1**

Original Cuda fp16 data size and format.

enum **NppiInterpolationMode**

Filtering methods.

Values:

enumerator **NPPI_INTER_UNDEFINED**

Undefined filtering interpolation mode.

enumerator **NPPI_INTER_NN**

Nearest neighbor filtering.

enumerator **NPPI_INTER_LINEAR**

Linear interpolation.

enumerator **NPPI_INTER_CUBIC**

Cubic interpolation.

enumerator **NPPI_INTER_CUBIC2P_BSPLINE**

Two-parameter cubic filter (B=1, C=0)

enumerator **NPPI_INTER_CUBIC2P_CATMULLROM**

Two-parameter cubic filter (B=0, C=1/2)

enumerator **NPPI_INTER_CUBIC2P_B05C03**

Two-parameter cubic filter (B=1/2, C=3/10)

enumerator **NPPI_INTER_SUPER**

Super sampling.

enumerator **NPPI_INTER_LANCZOS**

Lanczos filtering.

enumerator **NPPI_INTER_LANCZOS3_ADVANCED**

Generic Lanczos filtering with order 3.

enumerator **NPPI_SMOOTH_EDGE**

Smooth edge filtering.

enum **NppiBayerGridPosition**

Bayer Grid Position Registration.

Values:

enumerator **NPPI_BAYER_BGGR**

Default registration position BGGR.

enumerator **NPPI_BAYER_RGGB**

Registration position RGGB.

enumerator **NPPI_BAYER_GBRG**

Registration position GBRG.

enumerator **NPPI_BAYER_GRBG**

Registration position GRBG.

enum **NppiMaskSize**

Fixed filter-kernel sizes.

Values:

enumerator **NPP_MASK_SIZE_1_X_3**

1 X 3 filter mask size.

enumerator **NPP_MASK_SIZE_1_X_5**

1 X 5 filter mask size.

enumerator **NPP_MASK_SIZE_3_X_1**

3 X 1 filter mask size, leaving space for more 1 X N type enum values.

enumerator **NPP_MASK_SIZE_5_X_1**

5 X 1 filter mask size.

enumerator **NPP_MASK_SIZE_3_X_3**

3 X 3 filter mask size, leaving space for more N X 1 type enum values.

enumerator **NPP_MASK_SIZE_5_X_5**

5 X 5 filter mask size.

enumerator **NPP_MASK_SIZE_7_X_7**

7 X 7 filter mask size.

enumerator **NPP_MASK_SIZE_9_X_9**

9 X 9 filter mask size.

enumerator **NPP_MASK_SIZE_11_X_11**

11 X 11 filter mask size.

enumerator **NPP_MASK_SIZE_13_X_13**

13 X 13 filter mask size.

enumerator **NPP_MASK_SIZE_15_X_15**

15 X 15 filter mask size.

enum **NppiDifferentialKernel**

Differential Filter types.

Values:

enumerator **NPP_FILTER_SOBEL**

Differential kernel filter type sobel.

enumerator **NPP_FILTER_SCHARR**

Differential kernel filter type scharr.

enum **NppStatus**

Error Status Codes.

Almost all NPP function return error-status information using these return codes. Negative return codes indicate errors, positive return codes indicate warnings, a return code of 0 indicates success.

Values:

enumerator **NPP_LIBRARY_VERSION_MISMATCH_ERROR**

Application and NPP library version mismatch.

enumerator **NPP_INVALID_HOST_POINTER_ERROR**

Invalid host memory pointer error.

enumerator **NPP_INVALID_DEVICE_POINTER_ERROR**

Invalid device memory pointer error.

enumerator **NPP_LUT_PALETTE_BITSIZE_ERROR**

Color look up table bitsize error.

enumerator **NPP_ZC_MODE_NOT_SUPPORTED_ERROR**

ZeroCrossing mode not supported error.

enumerator **NPP_NOT_SUFFICIENT_COMPUTE_CAPABILITY**

Not sufficient Cuda compute capability error.

enumerator **NPP_TEXTURE_BIND_ERROR**

Texture bind error.

enumerator **NPP_WRONG_INTERSECTION_ROI_ERROR**

Wrong intersection region of interest error.

enumerator **NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR**

Haar classifier pixel match error.

enumerator **NPP_MEMFREE_ERROR**

Memory free request error.

enumerator **NPP_MEMSET_ERROR**

Memory set request error.

enumerator **NPP_MEMCPY_ERROR**

Memory copy request error.

enumerator **NPP_ALIGNMENT_ERROR**

Memory alignment error.

enumerator **NPP_CUDA_KERNEL_EXECUTION_ERROR**

Cuda kernel execution error, most commonly Cuda kernel launch error.

enumerator **NPP_STREAM_CTX_ERROR**

Non CTX function support has been deprecated.

enumerator **NPP_ROUND_MODE_NOT_SUPPORTED_ERROR**

Unsupported round mode.

enumerator **NPP_QUALITY_INDEX_ERROR**

Image pixels are constant for quality index.

enumerator **NPP_RESIZE_NO_OPERATION_ERROR**

One of the output image dimensions is less than 1 pixel.

enumerator **NPP_OVERFLOW_ERROR**

Number overflows the upper or lower limit of the data type.

enumerator **NPP_NOT_EVEN_STEP_ERROR**

Step value is not pixel multiple.

enumerator **NPP_HISTOGRAM_NUMBER_OF_LEVELS_ERROR**

Number of levels for histogram is less than 2.

enumerator **NPP_LUT_NUMBER_OF_LEVELS_ERROR**

Number of levels for LUT is less than 2.

enumerator **NPP_ZERO_MASK_VALUE_ERROR**

All values of the mask are zero.

enumerator **NPP_CORRUPTED_DATA_ERROR**

Processed data is corrupted.

enumerator **NPP_CHANNEL_ORDER_ERROR**

Wrong order of the destination channels.

enumerator **NPP_DATA_TYPE_ERROR**

Data type is incorrect or not supported.

enumerator **NPP_QUADRANGLE_ERROR**

The quadrangle is nonconvex or degenerates into triangle, line or point.

enumerator **NPP_RECTANGLE_ERROR**

Size of the rectangle region is less than or equal to 1.

enumerator **NPP_COEFFICIENT_ERROR**

Unallowable values of the transformation coefficients.

enumerator **NPP_NUMBER_OF_CHANNELS_ERROR**

Bad or unsupported number of channels.

enumerator **NPP_COI_ERROR**

Channel of interest is not 1, 2, or 3.

enumerator **NPP_DIVISOR_ERROR**

Divisor is equal to zero.

enumerator **NPP_CHANNEL_ERROR**

Illegal channel index.

enumerator **NPP_STRIDE_ERROR**

Stride is less than the row length.

enumerator **NPP_ANCHOR_ERROR**

Anchor point is outside mask.

enumerator **NPP_MASK_SIZE_ERROR**

Lower bound is larger than upper bound.

enumerator **NPP_RESIZE_FACTOR_ERROR**

enumerator **NPP_INTERPOLATION_ERROR**

enumerator **NPP_MIRROR_FLIP_ERROR**

enumerator **NPP_MOMENT_00_ZERO_ERROR**

enumerator **NPP_THRESHOLD_NEGATIVE_LEVEL_ERROR**

enumerator **NPP_THRESHOLD_ERROR**

enumerator **NPP_FFT_FLAG_ERROR**

enumerator **NPP_FFT_ORDER_ERROR**

enumerator **NPP_STEP_ERROR**

Step is less or equal zero.

enumerator **NPP_NOT_SUPPORTED_MODE_ERROR**

Not supported mode error.

enumerator **NPP_CONTEXT_MATCH_ERROR**

enumerator **NPP_SCALE_RANGE_ERROR**

enumerator **NPP_OUT_OFF_RANGE_ERROR**

enumerator **NPP_DIVIDE_BY_ZERO_ERROR**

enumerator **NPP_MEMORY_ALLOCATION_ERR**

enumerator **NPP_NULL_POINTER_ERROR**

enumerator **NPP_RANGE_ERROR**

enumerator **NPP_SIZE_ERROR**

enumerator **NPP_BAD_ARGUMENT_ERROR**

enumerator **NPP_NO_MEMORY_ERROR**

enumerator **NPP_NOT_IMPLEMENTED_ERROR**

enumerator **NPP_ERROR**

enumerator **NPP_ERROR_RESERVED**

enumerator **NPP_NO_ERROR**

Error free operation.

enumerator **NPP_SUCCESS**

Successful operation (same as NPP_NO_ERROR)

enumerator **NPP_NO_OPERATION_WARNING**

Indicates that no operation was performed.

enumerator **NPP_DIVIDE_BY_ZERO_WARNING**

Divisor is zero however does not terminate the execution.

enumerator **NPP_AFFINE_QUAD_INCORRECT_WARNING**

Indicates that the quadrangle passed to one of affine warping functions doesn't have necessary properties.

First 3 vertices are used, the fourth vertex discarded.

enumerator **NPP_WRONG_INTERSECTION_ROI_WARNING**

The given ROI has no intersection with either the source or destination ROI.

Thus no operation was performed.

enumerator **NPP_WRONG_INTERSECTION_QUAD_WARNING**

The given quadrangle has no intersection with either the source or destination ROI.

Thus no operation was performed.

enumerator **NPP_DOUBLE_SIZE_WARNING**

Image size isn't multiple of two.

Indicates that in case of 422/411/420 sampling the ROI width/height was modified for proper processing.

enumerator **NPP_MISALIGNED_DST_ROI_WARNING**

Speed reduction due to uncoalesced memory accesses warning.

struct **NppLibraryVersion**

NPPLibraryVersion This struct contains the NPP Library Version information.

Public Members

int **major**

Major version number.

int **minor**

Minor version number.

int **build**

Build number.

This reflects the nightly build this release was made from.

typedef unsigned char **Npp8u**

8-bit unsigned chars

typedef signed char **Npp8s**

8-bit signed chars

typedef unsigned short **Npp16u**

16-bit unsigned integers

typedef short **Npp16s**

16-bit signed integers

typedef unsigned int **Npp32u**

32-bit unsigned integers

typedef int **Npp32s**

32-bit signed integers

typedef unsigned long long **Npp64u**

64-bit unsigned integers

typedef long long **Npp64s**

64-bit signed integers

typedef float **Npp32f**

32-bit (IEEE) floating-point numbers

typedef double **Npp64f**

64-bit floating-point numbers

struct **Npp8uc**

Complex Number This struct represents an unsigned char complex number.

Public Members*Npp8u re*

Real part.

Npp8u im

Imaginary part.

struct **Npp16uc**

Complex Number This struct represents an unsigned short complex number.

Public Members*Npp16u re*

Real part.

Npp16u im

Imaginary part.

struct **Npp16sc**

Complex Number This struct represents a short complex number.

Public Members*Npp16s re*

Real part.

Npp16s im

Imaginary part.

struct **Npp32uc**

Complex Number This struct represents an unsigned int complex number.

Public Members*Npp32u re*

Real part.

Npp32u im

Imaginary part.

struct **Npp32sc**

Complex Number This struct represents a signed int complex number.

Public Members

Npp32s **re**

Real part.

Npp32s **im**

Imaginary part.

struct **Npp32fc**

Complex Number This struct represents a single floating-point complex number.

Public Members

Npp32f **re**

Real part.

Npp32f **im**

Imaginary part.

struct **Npp64sc**

Complex Number This struct represents a long long complex number.

Public Members

Npp64s **re**

Real part.

Npp64s **im**

Imaginary part.

struct **Npp64fc**

Complex Number This struct represents a double floating-point complex number.

Public Members

Npp64f **re**

Real part.

Npp64f **im**

Imaginary part.

enum NppDataType

Data types for nppiPlus functions.

Values:

enumerator **NPP_8U**

8-bit unsigned integer data type

enumerator **NPP_8S**

8-bit signed integer data type

enumerator **NPP_16U**

16-bit unsigned integer data type

enumerator **NPP_16S**

16-bit signed integer data type

enumerator **NPP_32U**

32-bit unsigned integer data type

enumerator **NPP_32S**

32-bit signed integer data type

enumerator **NPP_64U**

64-bit unsigned integer data type

enumerator **NPP_64S**

64-bit signed integer data type

enumerator **NPP_16F**

16-bit original Cuda floating point data type

enumerator **NPP_32F**

32-bit floating point data type

enumerator **NPP_64F**

64-bit double precision floating point data type

enum NppiChannels

Image channel counts for nppiPlus functions.

Values:

enumerator **NPP_CH_1**

Single channel per pixel data.

enumerator **NPP_CH_2**

Two channel per pixel data.

enumerator **NPP_CH_3**

Three channel per pixel data.

enumerator **NPP_CH_4**

Four channel per pixel data.

enumerator **NPP_CH_A4**

Four channel per pixel data with alpha.

enumerator **NPP_CH_P2**

Two channel single channel per plane pixel data.

enumerator **NPP_CH_P3**

Three channel single channel per plane pixel data.

enumerator **NPP_CH_P4**

Four channel single channel per plane pixel data.

NPP_MIN_8U

Minimum 8-bit unsigned integer.

NPP_MAX_8U

Maximum 8-bit unsigned integer.

NPP_MIN_16U

Minimum 16-bit unsigned integer.

NPP_MAX_16U

Maximum 16-bit unsigned integer.

NPP_MIN_32U

Minimum 32-bit unsigned integer.

NPP_MAX_32U

Maximum 32-bit unsigned integer.

NPP_MIN_64U

Minimum 64-bit unsigned integer.

NPP_MAX_64U

Maximum 64-bit unsigned integer.

NPP_MIN_8S

Minimum 8-bit signed integer.

NPP_MAX_8S

Maximum 8-bit signed integer.

NPP_MIN_16S

Minimum 16-bit signed integer.

NPP_MAX_16S

Maximum 16-bit signed integer.

NPP_MIN_32S

Minimum 32-bit signed integer.

NPP_MAX_32S

Maximum 32-bit signed integer.

NPP_MIN_64S

Minimum 64-bit signed integer.

NPP_MAX_64S

Minimum 64-bit signed integer.

NPP_MINABS_32F

Smallest positive 32-bit floating point value.

NPP_MAXABS_32F

Largest positive 32-bit floating point value.

NPP_MINABS_64F

Smallest positive 64-bit floating point value.

NPP_MAXABS_64F

Largest positive 64-bit floating point value.

struct NppiPoint

2D Point

Public Members

int **x**
x-coordinate.

int **y**
y-coordinate.

struct **NppiPoint32f**
2D Npp32f Point

Public Members

Npp32f **x**
x-coordinate.

Npp32f **y**
y-coordinate.

struct **NppiPoint64f**
2D Npp64f Point

Public Members

Npp64f **x**
x-coordinate.

Npp64f **y**
y-coordinate.

struct **NppPointPolar**
2D Polar Point

Public Members

Npp32f **rho**
Polar rho value.

Npp32f **theta**
Polar theta value.

struct **NppiSize**
2D Size This struct typically represents the size of a rectangular region in two space.

Public Membersint **width**

Rectangle width.

int **height**

Rectangle height.

struct **NppiRect**

2D Rectangle This struct contains position and size information of a rectangle in two space.

The rectangle's position is usually signified by the coordinate of its upper-left corner.

Public Membersint **x**

x-coordinate of upper left corner (lowest memory address).

int **y**

y-coordinate of upper left corner (lowest memory address).

int **width**

Rectangle width.

int **height**

Rectangle height.

enum **NppiAxis**

nppiMirror direction controls

*Values:*enumerator **NPP_HORIZONTAL_AXIS**

Flip around horizontal axis in mirror function.

enumerator **NPP_VERTICAL_AXIS**

Flip around vertical axis in mirror function.

enumerator **NPP_BOTH_AXIS**

Flip around both axes in mirror function.

enum **NppCmpOp**

Pixel comparison control values.

Values:

enumerator **NPP_CMP_LESS**

Threshold test for less than.

enumerator **NPP_CMP_LESS_EQ**

Threshold test for less than or equal.

enumerator **NPP_CMP_EQ**

Threshold test for equal.

enumerator **NPP_CMP_GREATER_EQ**

Threshold test for greater than or equal.

enumerator **NPP_CMP_GREATER**

Threshold test for greater than.

enum **NppRoundMode**

Rounding Modes.

The enumerated rounding modes are used by a large number of NPP primitives to allow the user to specify the method by which fractional values are converted to integer values. Also see *Round-[ing Modes](#)*.

For NPP release 5.5 new names for the three rounding modes are introduced that are based on the naming conventions for rounding modes set forth in the IEEE-754 floating-point standard. Developers are encouraged to use the new, longer names to be future proof as the legacy names will be deprecated in subsequent NPP releases.

Values:

enumerator **NPP_RND_NEAR**

Round to the nearest even integer.

All fractional numbers are rounded to their nearest integer. The ambiguous cases (i.e. <integer>.5) are rounded to the closest even integer. E.g.

- ▶ roundNear(0.5) = 0
- ▶ roundNear(0.6) = 1
- ▶ roundNear(1.5) = 2
- ▶ roundNear(-1.5) = -2

enumerator **NPP_ROUND_NEAREST_TIES_TO_EVEN**

Alias name for NPP_RND_NEAR.

enumerator **NPP_RND_FINANCIAL**

Round according to financial rule.

All fractional numbers are rounded to their nearest integer. The ambiguous cases (i.e. <integer>.5) are rounded away from zero. E.g.

- ▶ roundFinancial(0.4) = 0

- ▶ `roundFinancial(0.5) = 1`
- ▶ `roundFinancial(-1.5) = -2`

enumerator **NPP_ROUND_NEAREST_TIES_AWAY_FROM_ZERO**

Alias name for `NPP_RND_FINANCIAL`.

enumerator **NPP_RND_ZERO**

Round towards zero (truncation).

All fractional numbers of the form `<integer>.<decimals>` are truncated to `<integer>`.

- ▶ `roundZero(1.5) = 1`
- ▶ `roundZero(1.9) = 1`
- ▶ `roundZero(-2.5) = -2`

enumerator **NPP_ROUND_TOWARD_ZERO**

Alias name for `NPP_RND_ZERO`.

enum **NppiBorderType**

Supported image border modes.

Values:

enumerator **NPP_BORDER_UNDEFINED**

Image border type undefined.

enumerator **NPP_BORDER_NONE**

Image border type none.

enumerator **NPP_BORDER_CONSTANT**

Image border type constant value.

enumerator **NPP_BORDER_REPLICATE**

Image border type replicate border pixels.

enumerator **NPP_BORDER_WRAP**

Image border type wrap border pixels.

enumerator **NPP_BORDER_MIRROR**

Image border type mirror border pixels (IPP mirrored border / OpenCV `BORDER_REFLECT_101`): `gfedcb|abcdefgh|gfedcba` — edge pixel is NOT repeated.

enumerator **NPP_BORDER_MIRROR_REFL**

Image border type mirror with replication (IPP mirrored border with replication / OpenCV `BORDER_REFLECT`): `fedcba|abcdefgh|hgfedcba` — edge pixel IS repeated.

enum **NppHintAlgorithm**

Hints.

Values:

enumerator **NPP_ALG_HINT_NONE**

Hint none, currently these are all ignored.

enumerator **NPP_ALG_HINT_FAST**

Hint fast, currently these are all ignored.

enumerator **NPP_ALG_HINT_ACCURATE**

Hint accurate, currently these are all ignored.

enum **NppiAlphaOp**

Alpha composition mode controls.

Values:

enumerator **NPPI_OP_ALPHA_OVER**

Alpha composition over operation.

enumerator **NPPI_OP_ALPHA_IN**

Alpha composition in operation.

enumerator **NPPI_OP_ALPHA_OUT**

Alpha composition out operation.

enumerator **NPPI_OP_ALPHA_ATOP**

Alpha composition atop operation.

enumerator **NPPI_OP_ALPHA_XOR**

Alpha composition xor operation.

enumerator **NPPI_OP_ALPHA_PLUS**

Alpha composition plus operation.

enumerator **NPPI_OP_ALPHA_OVER_PREMUL**

Alpha composition over premultiply operation.

enumerator **NPPI_OP_ALPHA_IN_PREMUL**

Alpha composition in premultiply operation.

enumerator **NPPI_OP_ALPHA_OUT_PREMUL**

Alpha composition out premultiply operation.

enumerator **NPPI_OP_ALPHA_ATOP_PREMUL**
Alpha composition atop premultiply operation.

enumerator **NPPI_OP_ALPHA_XOR_PREMUL**
Alpha composition xor premultiply operation.

enumerator **NPPI_OP_ALPHA_PLUS_PREMUL**
Alpha composition plus premultiply operation.

enumerator **NPPI_OP_ALPHA_PREMUL**
Alpha composition premultiply operation.

struct **NppiHOGConfig**

The *NppiHOGConfig* structure defines the configuration parameters for the HOG descriptor:

Public Members

int **cellSize**
square cell size (pixels).

int **histogramBlockSize**
square histogram block size (pixels).

int **nHistogramBins**
required number of histogram bins.

NppiSize **detectionWindowSize**
detection window size (pixels).

NPP_HOG_MAX_CELL_SIZE
HOG Cell controls.
max horizontal/vertical pixel size of cell.

NPP_HOG_MAX_BLOCK_SIZE
max horizontal/vertical pixel size of block.

NPP_HOG_MAX_BINS_PER_CELL
max number of histogram bins.

NPP_HOG_MAX_CELLS_PER_DESCRIPTOR
max number of cells in a descriptor window.

NPP_HOG_MAX_OVERLAPPING_BLOCKS_PER_DESCRIPTOR
max number of overlapping blocks in a descriptor window.

NPP_HOG_MAX_DESCRIPTOR_LOCATIONS_PER_CALL

max number of descriptor window locations per function call.

struct **NppiHaarClassifier_32f**

Data structure for HaarClassifier_32f.

Public Members

int **numClassifiers**

number of classifiers.

Npp32s ***classifiers**

packed classifier data 40 bytes each.

size_t **classifierStep**

packed classifier byte step.

NppiSize **classifierSize**

packed classifier size.

Npp32s ***counterDevice**

counter device.

struct **NppiHaarBuffer**

Data structure for Haar buffer.

Public Members

int **haarBufferSize**

size of the buffer

Npp32s ***haarBuffer**

buffer

enum **NppsZCType**

Signal sign operations.

Values:

enumerator **nppZCR**

sign change

enumerator **nppZCXor**

sign change XOR

enumerator **nppZCC**
 sign change count_0

enum **NppiHuffmanTableType**
 HuffMan Table controls.
Values:

enumerator **nppiDCTable**
 DC Table.

enumerator **nppiACTable**
 AC Table.

enum **NppiNorm**
 Norm controls.
Values:

enumerator **nppiNormInf**
 maximum

enumerator **nppiNormL1**
 sum

enumerator **nppiNormL2**
 square root of sum of squares

struct **NppiConnectedRegion**
 Data structure of connected pixel region information.

Public Members

NppiRect **oBoundingBox**
 x, y, width, height == left, top, right, and bottom pixel coordinates

Npp32u **nConnectedPixelCount**
 total number of pixels in connected region

Npp32u **aSeedPixelValue[3]**
 original pixel value of seed pixel, 1 or 3 channel

struct **NppiImageDescriptor**
 General image descriptor.
 Defines the basic parameters of an image, including data pointer, step, and image size. This can be used by both source and destination images.

Public Members

void ***pData**

device memory pointer to the image

int **nStep**

step size

NppiSize **oSize**

width and height of the image

struct **NppiBufferDescriptor**

struct *NppiBufferDescriptor*

Public Members

void ***pData**

per image device memory pointer to the corresponding buffer

int **nBufferSize**

allocated buffer size

struct **NppiCompressedMarkerLabelsInfo**

Provides details of uniquely labeled pixel regions of interest returned by CompressedLabelMarkersUF function.

Public Members

Npp32u **nMarkerLabelPixelCount**

total number of pixels in this connected pixel region

Npp32u **nContourPixelCount**

total number of pixels in this connected pixel region contour

Npp32u **nContourPixelsFound**

total number of pixels in this connected pixel region contour found during geometry search

NppiPoint **oContourFirstPixelLocation**

image geometric x and y location of the first pixel in the contour

NppiRect **oMarkerLabelBoundingBox**

bounding box of this connected pixel region expressed as leftmostX, topmostY, rightmostX, and bottommostY

struct NppiContourBlockSegment

Provides details of contour pixel grid map location and association.

Public Members*Npp32u* **nMarkerLabelID**

this connected pixel region contour ID

Npp32u **nContourPixelCount**

total number of pixels in this connected pixel region contour

Npp32u **nContourStartingPixelOffset**

base offset of starting pixel in the overall contour pixel list

Npp32u **nSegmentNum**

relative block segment number within this particular contour ID

struct NppiContourPixelGeometryInfo

Provides contour (boundary) geometry info of uniquely labeled pixel regions returned by nppi-CompressedMarkerLabelsUInfo function in host memory in counterclockwise order relative to contour interiors.

Public Members*NppiPoint* **oContourOrderedGeometryLocation**

image geometry X and Y location in requested output order

NppiPoint **oContourPrevPixelLocation**

image geometry X and Y location of previous contour pixel

NppiPoint **oContourCenterPixelLocation**

image geometry X and Y location of center contour pixel

NppiPoint **oContourNextPixelLocation**

image geometry X and Y location of next contour pixel

Npp32s **nOrderIndex**

contour pixel counterclockwise order index in geometry list

Npp32s **nReverseOrderIndex**

contour pixel clockwise order index in geometry list

Npp32u **nFirstIndex**

index of first ordered contour pixel in this subgroup

Npp32u **nLastIndex**

index of last ordered contour pixel in this subgroup

Npp32u **nNextContourPixelIndex**

index of next ordered contour pixel in *NppiContourPixelGeometryInfo* list

Npp32u **nPrevContourPixelIndex**

index of previous ordered contour pixel in *NppiContourPixelGeometryInfo* list

Npp8u **nPixelAlreadyUsed**

this test pixel is has already been used

Npp8u **nAlreadyLinked**

this test pixel is already linked to center pixel

Npp8u **nAlreadyOutput**

this pixel has already been output in geometry list

Npp8u **nContourInteriorDirection**

direction of contour region interior

NPP_CONTOUR_DIRECTION_SOUTH_EAST

Provides contour (boundary) direction info of uniquely labeled pixel regions returned by CompressedLabelMarkersUF function.

Contour direction south east

NPP_CONTOUR_DIRECTION_SOUTH

Contour direction south.

NPP_CONTOUR_DIRECTION_SOUTH_WEST

Contour direction south west.

NPP_CONTOUR_DIRECTION_WEST

Contour direction west.

NPP_CONTOUR_DIRECTION_EAST

Contour direction east.

NPP_CONTOUR_DIRECTION_NORTH_EAST

Contour direction north east.

NPP_CONTOUR_DIRECTION_NORTH

Contour direction north.

NPP_CONTOUR_DIRECTION_NORTH_WEST

Contour direction north west.

NPP_CONTOUR_DIRECTION_ANY_NORTH

Provides contour (boundary) diagonal direction info of uniquely labeled pixel regions returned by CompressedLabelMarkersUF function.

Contour direction any north

NPP_CONTOUR_DIRECTION_ANY_WEST

Contour direction any west.

NPP_CONTOUR_DIRECTION_ANY_SOUTH

Contour direction any south.

NPP_CONTOUR_DIRECTION_ANY_EAST

Contour direction any east.

struct NppiContourPixelDirectionInfo

Data structure for contour pixel direction information.

Public Members***Npp32u* nMarkerLabelID**

MarkerLabelID of contour interior connected region.

***Npp8u* nContourDirectionCenterPixel**

provides current center contour pixel input and output direction info

***Npp8u* nContourInteriorDirectionCenterPixel**

provides current center contour pixel region interior direction info

***Npp8u* nConnected**

direction to directly connected contour pixels, 0 if not connected

***Npp8u* nGeometryInfoIsValid**

direction to directly connected contour pixels, 0 if not connected

***NppiContourPixelGeometryInfo* oContourPixelGeometryInfo**

Pixel geometry info structure.

***NppiPoint* nEast1**

Pixel coordinate values in each direction.

NppiPoint **nNorthEast1**

Pixel coordinate values in each direction.

NppiPoint **nNorth1**

Pixel coordinate values in each direction.

NppiPoint **nNorthWest1**

Pixel coordinate values in each direction.

NppiPoint **nWest1**

Pixel coordinate values in each direction.

NppiPoint **nSouthWest1**

Pixel coordinate values in each direction.

NppiPoint **nSouth1**

Pixel coordinate values in each direction.

NppiPoint **nSouthEast1**

Pixel coordinate values in each direction.

Npp8u **nTest1EastConnected**

East connected flag.

Npp8u **nTest1NorthEastConnected**

North east connected flag.

Npp8u **nTest1NorthConnected**

North connected flag.

Npp8u **nTest1NorthWestConnected**

North west connected flag.

Npp8u **nTest1WestConnected**

West connected flag.

Npp8u **nTest1SouthWestConnected**

South west connected flag.

Npp8u **nTest1SouthConnected**

South connected flag.

Npp8u **nTest1SouthEastConnected**

South east connected flag.

struct NppiContourTotalsInfo

Data structure for contour total counts.

Public Members**Npp32u nTotalImagePixelContourCount**

total number of contour pixels in image

Npp32u nLongestImageContourPixelCount

longest per contour pixel count in image

enum NppiWatershedSegmentBoundaryType

Provides control of the type of segment boundaries, if any, added to the image generated by the watershed segmentation function.

Values:

enumerator NPP_WATERSHED_SEGMENT_BOUNDARIES_NONE

Image watershed segment boundary type none.

enumerator NPP_WATERSHED_SEGMENT_BOUNDARIES_BLACK

Image watershed segment boundary type black.

enumerator NPP_WATERSHED_SEGMENT_BOUNDARIES_WHITE

Image watershed segment boundary type white.

enumerator NPP_WATERSHED_SEGMENT_BOUNDARIES_CONTRAST

Image watershed segment boundary type contrasting intensity.

enumerator NPP_WATERSHED_SEGMENT_BOUNDARIES_ONLY

Image watershed segment boundary type render boundaries only.

struct NppStreamContext

1.5.1. Application Managed Stream Context

NPP stream context structure must be filled in by application.

Populate all fields as shown below. Set hWorkspace = 0 to opt out of the optional workspace cache (pass a handle from nppiWorkspaceCreate() to enable it):

```
NppStreamContext ctx;
ctx.hStream = myStream;
cudaGetDevice(&ctx.nCudaDeviceId);
cudaDeviceProp props;
```

(continues on next page)

(continued from previous page)

```

cudaGetDeviceProperties(&props, ctx.nCudaDeviceId);
ctx.nMultiProcessorCount      = props.multiProcessorCount;
ctx.nMaxThreadsPerMultiProcessor = props.maxThreadsPerMultiProcessor;
ctx.nMaxThreadsPerBlock      = props.maxThreadsPerBlock;
ctx.nSharedMemPerBlock       = props.sharedMemPerBlock;
ctx.nCudaDevAttrComputeCapabilityMajor = props.major;
ctx.nCudaDevAttrComputeCapabilityMinor = props.minor;
cudaStreamGetFlags(myStream, &ctx.nStreamFlags);
ctx.hWorkspace = 0; // must be zero unless using nppiWorkspaceCreate()

```

Public Members

`cudaStream_t hStream`

From current Cuda stream ID.

`int nCudaDeviceId`

From `cudaGetDevice()`.

`int nMultiProcessorCount`

From `cudaGetDeviceProperties()`.

`int nMaxThreadsPerMultiProcessor`

From `cudaGetDeviceProperties()`.

`int nMaxThreadsPerBlock`

From `cudaGetDeviceProperties()`.

`size_t nSharedMemPerBlock`

From `cudaGetDeviceProperties()`.

`int nCudaDevAttrComputeCapabilityMajor`

From `cudaGetDeviceAttribute()`.

`int nCudaDevAttrComputeCapabilityMinor`

From `cudaGetDeviceAttribute()`.

`unsigned int nStreamFlags`

From `cudaStreamGetFlags()`.

`NppWorkspace hWorkspace`

Optional workspace handle; 0 = no workspace.

Set via `nppiWorkspaceCreate()`.

Npp32u `nReserved0`

Deprecated:

Use hWorkspace. Same type and storage, kept for source compatibility with existing code that sets nReserved0 = 0.

union *NppStreamContext*.[anonymous] [**anonymous**]

struct **NppiResizeBatchCXR**

NPP Batch Geometry Structure Definitions.

Public Members

const void ***pSrc**

source image device memory pointer.

int **nSrcStep**

source image byte count per row.

void ***pDst**

destination image device memory pointer.

int **nDstStep**

device image byte count per row.

struct **NppiResizeBatchROI_Advanced**

Data structure for variable ROI image batch resizing.

Public Members

NppiRect **oSrcRectROI**

per source image rectangle parameters.

NppiRect **oDstRectROI**

per destination image rectangle parameters.

struct **NppiMirrorBatchCXR**

Data structure for batched nppiMirrorBatch.

Public Members

const void ***pSrc**

source image device memory pointer, ignored for in-place versions.

int **nSrcStep**

source image byte count per row.

void ***pDst**

destination image device memory pointer.

int **nDstStep**

device image byte count per row.

struct **NppiWarpAffineBatchCXR**

Data structure for batched nppiWarpAffineBatch.

Public Members

const void ***pSrc**

source image device memory pointer.

int **nSrcStep**

source image byte count per row.

void ***pDst**

destination image device memory pointer.

int **nDstStep**

device image byte count per row.

Npp64f ***pCoeffs**

device memory pointer to the tranformation matrix with double precision floating-point coefficient values to be used for this image.

Npp64f **aTransformedCoeffs**[2][3]

FOR INTERNAL USE, DO NOT INITIALIZE.

struct **NppiWarpPerspectiveBatchCXR**

Data structure for batched nppiWarpPerspectiveBatch.

Public Members**const void *pSrc**

source image device memory pointer.

int nSrcStep

source image byte count per row.

void *pDst

destination image device memory pointer.

int nDstStep

device image byte count per row.

Npp64f *pCoeffs

device memory pointer to the transformation matrix with double precision floating-point coefficient values to be used for this image.

Npp64f aTransformedCoeffs[3][3]

FOR INTERNAL USE, DO NOT INITIALIZE.

struct NppiColorTwistBatchCXR

Data structure for batched nppiColorTwistBatch.

Public Members**const void *pSrc**

source image device memory pointer.

int nSrcStep

source image byte count per row.

void *pDst

destination image device memory pointer.

int nDstStep

device image byte count per row.

Npp32f *pTwist

device memory pointer to the color twist matrix with floating-point coefficient values to be used for this image.

1.6. Core NPP Functions

Basic functions for library management, in particular library version and device property query functions.

Functions

const *NppLibraryVersion* ***nppGetLibVersion**(void)

Get the NPP library version.

Returns

A struct containing separate values for major and minor revision and build number.

1.7. Image Arithmetic And Logical Operations

These functions can be found in the nppial library.

Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

1.7.1. Arithmetic Operations

1.7.1.1 Arithmetic Operations

The set of image processing arithmetic operations available in the library.

1.7.1.2 AddC

Adds a constant value to each pixel of an image.

Note: If you use one of the device constant versions of these functions and the function called immediately preceding that function generates that device constant you MUST either call `cudaStreamSynchronize()` or `cudaDeviceSynchronize()` before calling the device constant function.

Functions

NppStatus **nppiAddC_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **nConstant** – host memory Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstant** – pointer to device memory Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_8u_C1RSfs_Ctx**(const *Npp8u* nConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddDeviceC_8u_C1RSfs_Ctx(const Npp8u *pConstant, Npp8u *pSrcDst, int  
                                         nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
                                         NppStreamContext nppStreamCtx)
```

One 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – pointer to device memory Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddC_8u_C3RSfs_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u  
                                   aConstants[3], Npp8u *pDst, int nDstStep, NppiSize  
                                   oSizeROI, int nScaleFactor, NppStreamContext  
                                   nppStreamCtx)
```

Three 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddDeviceC_8u_C3RSfs_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u  
                                           *pConstants, Npp8u *pDst, int nDstStep, NppiSize  
                                           oSizeROI, int nScaleFactor, NppStreamContext  
                                           nppStreamCtx)
```

Three 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_8u_C3IRSfs_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel 8-bit unsigned char in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_8u_C3IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel 8-bit unsigned char in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddC_8u_AC4RSfs_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u
                                   aConstants[3], Npp8u *pDst, int nDstStep, NppiSize
                                   oSizeROI, int nScaleFactor, NppStreamContext
                                   nppStreamCtx)
```

Four 8-bit unsigned char channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddDeviceC_8u_AC4RSfs_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u
                                           *pConstants, Npp8u *pDst, int nDstStep, NppiSize
                                           oSizeROI, int nScaleFactor, NppStreamContext
                                           nppStreamCtx)
```

Four 8-bit unsigned char channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddC_8u_AC4IRSfs_Ctx(const Npp8u aConstants[3], Npp8u *pSrcDst, int
                                     nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,
                                     NppStreamContext nppStreamCtx)
```

Four 8-bit unsigned char channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_8u_AC4IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_8u_C4IRSfs_Ctx**(const *Npp8u* aConstants[4], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_8u_C4IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* nConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16u_C1IRSfs_Ctx**(const *Npp16u* nConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- **nConstant** – host memory constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16u_C1IRSfs_Ctx**(const *Npp16u* *pConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- **pConstant** – device memory constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16u_C3IRSfs_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16u_C3IRSfs_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16u_AC4IRSfs_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16u_AC4IRSfs_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*

- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16u_C4IRSfs_Ctx**(const *Npp16u* aConstants[4], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16u_C4IRSfs_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* nConstant, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstant, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16s_C1IRSfs_Ctx**(const *Npp16s* nConstant, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16s_C1IRSfs_Ctx**(const *Npp16s* *pConstant, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* aConstants[3], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstants, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16s_C3IRSfs_Ctx**(const *Npp16s* aConstants[3], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16s_C3IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddC_16s_AC4RSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s
                                     aConstants[3], Npp16s *pDst, int nDstStep, NppiSize
                                     oSizeROI, int nScaleFactor, NppStreamContext
                                     nppStreamCtx)
```

Four 16-bit signed short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddDeviceC_16s_AC4RSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const
                                             Npp16s *pConstants, Npp16s *pDst, int nDstStep,
                                             NppiSize oSizeROI, int nScaleFactor,
                                             NppStreamContext nppStreamCtx)
```

Four 16-bit signed short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddC_16s_AC4IRSfs_Ctx(const Npp16s aConstants[3], Npp16s *pSrcDst, int
                                         nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,
                                         NppStreamContext nppStreamCtx)
```

Four 16-bit signed short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16s_AC4IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* aConstants[4], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstants, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image add constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16s_C4IRSfs_Ctx**(const *Npp16s* aConstants[4], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16s_C4IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* nConstant, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16sc_C1IRSfs_Ctx**(const *Npp16sc* nConstant, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16sc_C3RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* aConstants[3], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16sc_C3IRSfs_Ctx**(const *Npp16sc* aConstants[3], *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16sc_AC4RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* aConstants[3], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16sc_AC4IRSfs_Ctx**(const *Npp16sc* aConstants[3], *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* nConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstant** – device memory constant.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32s_C1RSfs_Ctx**(const *Npp32s* nConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

Parameters

- **nConstant** – host memory constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32s_C1IRSfs_Ctx**(const *Npp32s* *pConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pConstants, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32s_C3IRSfs_Ctx**(const *Npp32s* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32s_C3IRSfs_Ctx**(const *Npp32s* *pConstants, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

Parameters

- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32sc_C1RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* nConstant, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32sc_C1IRSfs_Ctx**(const *Npp32sc* nConstant, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32sc_C3RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* aConstants[3], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32sc_C3IRSfs_Ctx**(const *Npp32sc* aConstants[3], *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32sc_AC4RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* aConstants[3], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32sc_AC4IRSfs_Ctx**(const *Npp32sc* aConstants[3], *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* nConstant, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory 32-bit floating point constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstant, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstant** – device memory 32-bit floating point constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16f_C1IR_Ctx**(const *Npp32f* nConstant, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image add constant.

Parameters

- ▶ **nConstant** – host memory 32-bit floating point constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16f_C1IR_Ctx**(const *Npp32f* *pConstant, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image add constant.

Parameters

- ▶ **pConstant** – device memory 32-bit floating point constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16f_C3R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddDeviceC_16f_C3R_Ctx(const Npp16f *pSrc1, int nSrc1Step, const Npp32f
                                     *pConstants, Npp16f *pDst, int nDstStep, NppiSize
                                     oSizeROI, NppStreamContext nppStreamCtx)
```

Three 16-bit floating point channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddC_16f_C3IR_Ctx(const Npp32f aConstants[3], Npp16f *pSrcDst, int
                                nSrcDstStep, NppiSize oSizeROI, NppStreamContext
                                nppStreamCtx)
```

Three 16-bit floating point channel in place image add constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAddDeviceC_16f_C3IR_Ctx(const Npp32f *pConstants, Npp16f *pSrcDst, int
                                       nSrcDstStep, NppiSize oSizeROI, NppStreamContext
                                       nppStreamCtx)
```

Three 16-bit floating point channel in place image add constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16f_C4R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[4], *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image add constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16f_C4R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image add constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_16f_C4IR_Ctx**(const *Npp32f* aConstants[4], *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image add constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_16f_C4IR_Ctx**(const *Npp32f* *pConstants, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image add constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* nConstant, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstant, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32f_C1IR_Ctx**(const *Npp32f* nConstant, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image add constant.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32f_C1IR_Ctx**(const *Npp32f* *pConstant, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image add constant.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32f_C3IR_Ctx**(const *Npp32f* aConstants[3], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image add constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32f_C3IR_Ctx**(const *Npp32f* *pConstants, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image add constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32f_AC4IR_Ctx**(const *Npp32f* aConstants[3], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image add constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32f_AC4IR_Ctx**(const *Npp32f* *pConstants, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image add constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[4], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*

- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image add constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32f_C4IR_Ctx**(const *Npp32f* aConstants[4], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image add constant.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddDeviceC_32f_C4IR_Ctx**(const *Npp32f* *pConstants, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image add constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32fc_C1R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* nConstant, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32fc_C1IR_Ctx**(const *Npp32fc* nConstant, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32fc_C3R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[3], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32fc_C3IR_Ctx**(const *Npp32fc* aConstants[3], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32fc_AC4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[3], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image add constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32fc_AC4IR_Ctx**(const *Npp32fc* aConstants[3], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image add constant.

Parameters

- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32fc_C4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[4], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddC_32fc_C4IR_Ctx**(const *Npp32fc* aConstants[4], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

Parameters

- **aConstants** – fixed size array of host memory constant values, one per channel.

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.3 MulC

Multiplies each pixel of an image by a constant value.

Note: If you use one of the device constant versions of these functions and the function called immediately preceding that function generates that device constant you MUST either call `cudaStreamSynchronize()` or `cudaDeviceSynchronize()` before calling the device constant function.

Functions

NppStatus **nppiMulC_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*

- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_8u_C1IRSfs_Ctx**(const *Npp8u* nConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_8u_C1IRSfs_Ctx**(const *Npp8u* *pConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulC_8u_C3RSfs_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u
                                aConstants[3], Npp8u *pDst, int nDstStep, NppiSize
                                oSizeROI, int nScaleFactor, NppStreamContext
                                nppStreamCtx)
```

Three 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulDeviceC_8u_C3RSfs_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u
                                         *pConstants, Npp8u *pDst, int nDstStep, NppiSize
                                         oSizeROI, int nScaleFactor, NppStreamContext
                                         nppStreamCtx)
```

Three 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulC_8u_C3IRSfs_Ctx(const Npp8u aConstants[3], Npp8u *pSrcDst, int
                                   nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,
                                   NppStreamContext nppStreamCtx)
```

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_8u_C3IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_8u_AC4IRSfs_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_8u_AC4IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.

- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_8u_C4IRSfs_Ctx**(const *Npp8u* aConstants[4], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_8u_C4IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* nConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulDeviceC_16u_C1RSfs_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pConstant, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor, NppStreamContext nppStreamCtx)
```

One 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulC_16u_C1IRSfs_Ctx(const Npp16u nConstant, Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor, NppStreamContext nppStreamCtx)
```

One 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16u_C1IRSfs_Ctx**(const *Npp16u* *pConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulDeviceC_16u_C3RSfs_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u
                                         *pConstants, Npp16u *pDst, int nDstStep, NppiSize
                                         oSizeROI, int nScaleFactor, NppStreamContext
                                         nppStreamCtx)
```

Three 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulC_16u_C3IRSfs_Ctx(const Npp16u aConstants[3], Npp16u *pSrcDst, int
                                     nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,
                                     NppStreamContext nppStreamCtx)
```

Three 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulDeviceC_16u_C3IRSfs_Ctx(const Npp16u *pConstants, Npp16u *pSrcDst, int
                                         nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,
                                         NppStreamContext nppStreamCtx)
```

Three 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16u_AC4IRSfs_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16u_AC4IRSfs_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16u_C4IRSfs_Ctx**(const *Npp16u* aConstants[4], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16u_C4IRSfs_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* nConstant, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstant, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16s_C1RSfs_Ctx**(const *Npp16s* nConstant, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16s_C1RSfs_Ctx**(const *Npp16s* *pConstant, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.

- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulC_16s_C3RSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s
aConstants[3], Npp16s *pDst, int nDstStep, NppiSize
oSizeROI, int nScaleFactor, NppStreamContext
nppStreamCtx)
```

Three 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulDeviceC_16s_C3RSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s
*pConstants, Npp16s *pDst, int nDstStep, NppiSize
oSizeROI, int nScaleFactor, NppStreamContext
nppStreamCtx)
```

Three 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16s_C3IRSfs_Ctx**(const *Npp16s* aConstants[3], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16s_C3IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16s_AC4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* aConstants[3], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16s_AC4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstants, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16s_AC4IRSfs_Ctx**(const *Npp16s* aConstants[3], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16s_AC4IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* aConstants[4], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstants, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16s_C4IRSfs_Ctx**(const *Npp16s* aConstants[4], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16s_C4IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* nConstant, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **nConstant** – host memory constant.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16sc_C1IRSfs_Ctx**(const *Npp16sc* nConstant, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- **nConstant** – host memory constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16sc_C3RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* aConstants[3], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16sc_C3IRSfs_Ctx**(const *Npp16sc* aConstants[3], *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16sc_AC4RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* aConstants[3], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16sc_AC4IRSfs_Ctx**(const *Npp16sc* aConstants[3], *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* nConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32s_C1IRSfs_Ctx**(const *Npp32s* nConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_32s_C1IRSfs_Ctx**(const *Npp32s* *pConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pConstants, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulC_32s_C3IRSfs_Ctx(const Npp32s aConstants[3], Npp32s *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

Three 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulDeviceC_32s_C3IRSfs_Ctx(const Npp32s *pConstants, Npp32s *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

Three 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulC_32sc_C1RSfs_Ctx(const Npp32sc *pSrc1, int nSrc1Step, const Npp32sc  
nConstant, Npp32sc *pDst, int nDstStep, NppiSize  
oSizeROI, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.

- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32sc_C1IRSfs_Ctx**(const *Npp32sc* nConstant, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- **nConstant** – host memory constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32sc_C3RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* aConstants[3], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32sc_C3IRSfs_Ctx**(const *Npp32sc* aConstants[3], *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32sc_AC4RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* aConstants[3], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32sc_AC4IRSfs_Ctx**(const *Npp32sc* aConstants[3], *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* nConstant, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – 32-bit floating point host memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstant, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – 32-bit floating point device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16f_C1IR_Ctx**(const *Npp32f* nConstant, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **nConstant** – 32-bit floating point host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16f_C1IR_Ctx**(const *Npp32f* *pConstant, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **pConstant** – 32-bit floating point device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16f_C3R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16f_C3R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16f_C3IR_Ctx**(const *Npp32f* aConstants[3], *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16f_C3IR_Ctx**(const *Npp32f* *pConstants, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16f_C4R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[4], *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image multiply by constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16f_C4R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image multiply by constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_16f_C4IR_Ctx**(const *Npp32f* aConstants[4], *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_16f_C4IR_Ctx**(const *Npp32f* *pConstants, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* nConstant, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstant, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32f_C1IR_Ctx**(const *Npp32f* nConstant, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_32f_C1IR_Ctx**(const *Npp32f* *pConstant, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32f_C3IR_Ctx**(const *Npp32f* aConstants[3], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulDeviceC_32f_C3IR_Ctx(const Npp32f *pConstants, Npp32f *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
nppStreamCtx)
```

Three 32-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulC_32f_AC4R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f  
aConstants[3], Npp32f *pDst, int nDstStep, NppiSize  
oSizeROI, NppStreamContext nppStreamCtx)
```

Four 32-bit floating point channel with unmodified alpha image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulDeviceC_32f_AC4R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f  
*pConstants, Npp32f *pDst, int nDstStep, NppiSize  
oSizeROI, NppStreamContext nppStreamCtx)
```

Four 32-bit floating point channel with unmodified alpha image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32f_AC4IR_Ctx**(const *Npp32f* aConstants[3], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image multiply by constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_32f_AC4IR_Ctx**(const *Npp32f* *pConstants, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image multiply by constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[4], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*

- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image multiply by constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32f_C4IR_Ctx**(const *Npp32f* aConstants[4], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image multiply by constant.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceC_32f_C4IR_Ctx**(const *Npp32f* *pConstants, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image multiply by constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32fc_C1R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* nConstant, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32fc_C1IR_Ctx**(const *Npp32fc* nConstant, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32fc_C3R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[3], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32fc_C3IR_Ctx**(const *Npp32fc* aConstants[3], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32fc_AC4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[3], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image multiply by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32fc_AC4IR_Ctx**(const *Npp32fc* aConstants[3], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image multiply by constant.

Parameters

- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32fc_C4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[4], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulC_32fc_C4IR_Ctx**(const *Npp32fc* aConstants[4], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

Parameters

- **aConstants** – fixed size array of host memory constant values, one per channel.

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.4 MulCScale

Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.

Note: If you use one of the device constant versions of these functions and the function called immediately preceding that function generates that device constant you MUST either call `cudaStreamSynchronize()` or `cudaDeviceSynchronize()` before calling the device constant function.

Functions

NppStatus **nppiMulCScale_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.

- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_8u_C1IR_Ctx**(const *Npp8u* nConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

Parameters

- **nConstant** – host memory constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_8u_C1IR_Ctx**(const *Npp8u* *pConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

Parameters

- **pConstant** – device memory constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

Parameters

- **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_8u_C3IR_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_8u_C3IR_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_8u_AC4IR_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale and scale by max bit width value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_8u_AC4IR_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale and scale by max bit width value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_8u_C4IR_Ctx**(const *Npp8u* aConstants[4], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_8u_C4IR_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* nConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_16u_C1IR_Ctx**(const *Npp16u* nConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_16u_C1IR_Ctx**(const *Npp16u* *pConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_16u_C3IR_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_16u_C3IR_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_16u_AC4IR_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_16u_AC4IR_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulCScale_16u_C4IR_Ctx**(const *Npp16u* aConstants[4], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulDeviceCScale_16u_C4IR_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.5 SubC

Subtracts a constant value from each pixel of an image.

Note: If you use one of the device constant versions of these functions and the function called immediately preceding that function generates that device constant you MUST either call `cudaStreamSynchronize()` or `cudaDeviceSynchronize()` before calling the device constant function.

Functions

NppStatus **nppiSubC_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiSubDeviceC_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiSubC_8u_C1RSfs_Ctx**(const *Npp8u* nConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiSubDeviceC_8u_C1RSfs_Ctx**(const *Npp8u* *pConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSubC_8u_C3RSfs_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u  
aConstants[3], Npp8u *pDst, int nDstStep, NppiSize  
oSizeROI, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Three 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSubDeviceC_8u_C3RSfs_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u  
*pConstants, Npp8u *pDst, int nDstStep, NppiSize  
oSizeROI, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Three 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_8u_C3IRSfs_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel 8-bit unsigned char in place image subtract constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_8u_C3IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel 8-bit unsigned char in place image subtract constant, scale, then clamp to saturated value.

Parameters

- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_8u_AC4IRSfs_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_8u_AC4IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters

- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSubC_8u_C4IRSfs_Ctx(const Npp8u aConstants[4], Npp8u *pSrcDst, int  
                                nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
                                NppStreamContext nppStreamCtx)
```

Four 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSubDeviceC_8u_C4IRSfs_Ctx(const Npp8u *pConstants, Npp8u *pSrcDst, int  
                                nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
                                NppStreamContext nppStreamCtx)
```

Four 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* nConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSubC_16u_C1IRSfs_Ctx(const Npp16u nConstant, Npp16u *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

One 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSubDeviceC_16u_C1IRSfs_Ctx(const Npp16u *pConstant, Npp16u *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

One 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSubC_16u_C3RSfs_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u  
aConstants[3], Npp16u *pDst, int nDstStep, NppiSize  
oSizeROI, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Three 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16u_C3IRSfs_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16u_C3IRSfs_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16u_AC4IRSfs_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16u_AC4IRSfs_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16u_C4IRSfs_Ctx**(const *Npp16u* aConstants[4], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16u_C4IRSfs_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* nConstant, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstant, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16s_C1IRSfs_Ctx**(const *Npp16s* nConstant, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16s_C1IRSfs_Ctx**(const *Npp16s* *pConstant, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* aConstants[3], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstants, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16s_C3IRSfs_Ctx**(const *Npp16s* aConstants[3], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16s_C3IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16s_AC4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* aConstants[3], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16s_AC4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstants, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16s_AC4IRSfs_Ctx**(const *Npp16s* aConstants[3], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16s_AC4IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* aConstants[4], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstants, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16s_C4IRSfs_Ctx**(const *Npp16s* aConstants[4], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16s_C4IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* nConstant, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16sc_C1IRSfs_Ctx**(const *Npp16sc* nConstant, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16sc_C3RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* aConstants[3], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16sc_C3IRSfs_Ctx**(const *Npp16sc* aConstants[3], *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16sc_AC4RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* aConstants[3], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16sc_AC4IRSfs_Ctx**(const *Npp16sc* aConstants[3], *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* nConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstant** – device memory constant.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32s_C1IRSfs_Ctx**(const *Npp32s* nConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- **nConstant** – host memory constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_32s_C1IRSfs_Ctx**(const *Npp32s* *pConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pConstants, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32s_C3IRSfs_Ctx**(const *Npp32s* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_32s_C3IRSfs_Ctx**(const *Npp32s* *pConstants, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32sc_C1RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* nConstant, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32sc_C1IRSfs_Ctx**(const *Npp32sc* nConstant, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32sc_C3RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* aConstants[3], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32sc_C3IRSfs_Ctx**(const *Npp32sc* aConstants[3], *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32sc_AC4RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* aConstants[3], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32sc_AC4IRSfs_Ctx**(const *Npp32sc* aConstants[3], *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* nConstant, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory 32-bit floating point constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstant, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstant** – device memory 32-bit floating point constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16f_C1IR_Ctx**(const *Npp32f* nConstant, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image subtract constant.

Parameters

- **nConstant** – host memory 32-bit floating point constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16f_C1IR_Ctx**(const *Npp32f* *pConstant, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image subtract constant.

Parameters

- **pConstant** – device memory 32-bit floating point constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16f_C3R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image subtract constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16f_C3R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16f_C3IR_Ctx**(const *Npp32f* aConstants[3], *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image subtract constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16f_C3IR_Ctx**(const *Npp32f* *pConstants, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image subtract constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_16f_C4R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[4], *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_16f_C4R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSubC_16f_C4IR_Ctx(const Npp32f aConstants[4], Npp16f *pSrcDst, int  
                                nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
                                nppStreamCtx)
```

Four 16-bit floating point channel in place image subtract constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSubDeviceC_16f_C4IR_Ctx(const Npp32f *pConstants, Npp16f *pSrcDst, int  
                                         nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
                                         nppStreamCtx)
```

Four 16-bit floating point channel in place image subtract constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSubC_32f_C1R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f nConstant,  
                                Npp32f *pDst, int nDstStep, NppiSize oSizeROI,  
                                NppStreamContext nppStreamCtx)
```

One 32-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiSubDeviceC_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstant, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiSubC_32f_C1R_Ctx**(const *Npp32f* nConstant, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image subtract constant.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiSubDeviceC_32f_C1IR_Ctx**(const *Npp32f* *pConstant, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image subtract constant.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32f_C3IR_Ctx**(const *Npp32f* aConstants[3], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image subtract constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*

- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_32f_C3IR_Ctx**(const *Npp32f* *pConstants, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image subtract constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32f_AC4IR_Ctx**(const *Npp32f* aConstants[3], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image subtract constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_32f_AC4IR_Ctx**(const *Npp32f* *pConstants, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image subtract constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[4], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32f_C4IR_Ctx**(const *Npp32f* aConstants[4], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image subtract constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubDeviceC_32f_C4IR_Ctx**(const *Npp32f* *pConstants, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image subtract constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32fc_C1R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* nConstant, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32fc_C1IR_Ctx**(const *Npp32fc* nConstant, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32fc_C3R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[3], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32fc_C3IR_Ctx**(const *Npp32fc* aConstants[3], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32fc_AC4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[3], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image subtract constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32fc_AC4IR_Ctx**(const *Npp32fc* aConstants[3], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image subtract constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32fc_C4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[4], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSubC_32fc_C4IR_Ctx**(const *Npp32fc* aConstants[4], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.6 DivC

Divides each pixel of an image by a constant value.

Note: If you use one of the device constant versions of these functions and the function called immediately preceding that function generates that device constant you MUST either call `cudaStreamSynchronize()` or `cudaDeviceSynchronize()` before calling the device constant function.

Functions

NppStatus **nppiDivC_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivDeviceC_8u_C1RSfs_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u
                                         *pConstant, Npp8u *pDst, int nDstStep, NppiSize
                                         oSizeROI, int nScaleFactor, NppStreamContext
                                         nppStreamCtx)
```

One 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivC_8u_C1RSfs_Ctx(const Npp8u nConstant, Npp8u *pSrcDst, int nSrcDstStep,
                                   NppiSize oSizeROI, int nScaleFactor, NppStreamContext
                                   nppStreamCtx)
```

One 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivDeviceC_8u_C1RSfs_Ctx(const Npp8u *pConstant, Npp8u *pSrcDst, int
                                         nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,
                                         NppStreamContext nppStreamCtx)
```

One 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.

- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_8u_C3IRSfs_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel 8-bit unsigned char in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_8u_C3IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel 8-bit unsigned char in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_8u_AC4IRSfs_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_8u_AC4IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters

- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_8u_C4IRSfs_Ctx**(const *Npp8u* aConstants[4], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_8u_C4IRSfs_Ctx**(const *Npp8u* *pConstants, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* nConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **nConstant** – host memory constant.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstant** – device memory constant.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16u_C1IRSfs_Ctx**(const *Npp16u* nConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16u_C1IRSfs_Ctx**(const *Npp16u* *pConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16u_C3IRSfs_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16u_C3IRSfs_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16u_AC4IRSfs_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16u_AC4IRSfs_Ctx**(const *Npp16u* *pConstants, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pConstants, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivC_16u_C4IRSfs_Ctx(const Npp16u aConstants[4], Npp16u *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

Four 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivDeviceC_16u_C4IRSfs_Ctx(const Npp16u *pConstants, Npp16u *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

Four 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivC_16s_C1RSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s  
nConstant, Npp16s *pDst, int nDstStep, NppiSize oSizeROI,  
int nScaleFactor, NppStreamContext nppStreamCtx)
```

One 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.

- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstant, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstant** – device memory constant.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16s_C1IRSfs_Ctx**(const *Npp16s* nConstant, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- **nConstant** – host memory constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivDeviceC_16s_C1IRSfs_Ctx(const Npp16s *pConstant, Npp16s *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

One 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivC_16s_C3RSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s  
aConstants[3], Npp16s *pDst, int nDstStep, NppiSize  
oSizeROI, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Three 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivDeviceC_16s_C3RSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s  
*pConstants, Npp16s *pDst, int nDstStep, NppiSize  
oSizeROI, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Three 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16s_C3IRSfs_Ctx**(const *Npp16s* aConstants[3], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16s_C3IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivC_16s_AC4RSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s
                                   aConstants[3], Npp16s *pDst, int nDstStep, NppiSize
                                   oSizeROI, int nScaleFactor, NppStreamContext
                                   nppStreamCtx)
```

Four 16-bit signed short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivDeviceC_16s_AC4RSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const
                                           Npp16s *pConstants, Npp16s *pDst, int nDstStep,
                                           NppiSize oSizeROI, int nScaleFactor,
                                           NppStreamContext nppStreamCtx)
```

Four 16-bit signed short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivC_16s_AC4IRSfs_Ctx(const Npp16s aConstants[3], Npp16s *pSrcDst, int
                                       nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,
                                       NppStreamContext nppStreamCtx)
```

Four 16-bit signed short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16s_AC4IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* aConstants[4], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pConstants, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstants** – fixed size device memory array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16s_C4IRSfs_Ctx**(const *Npp16s* aConstants[4], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size host memory array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16s_C4IRSfs_Ctx**(const *Npp16s* *pConstants, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* nConstant, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16sc_C1IRSfs_Ctx**(const *Npp16sc* nConstant, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16sc_C3RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* aConstants[3], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16sc_C3IRSfs_Ctx**(const *Npp16sc* aConstants[3], *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16sc_AC4RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* aConstants[3], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16sc_AC4IRSfs_Ctx**(const *Npp16sc* aConstants[3], *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* nConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32s_C1RSfs**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* nConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor)

One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32s_C1RSfs_Ctx**(const *Npp32s* nConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32s_C1IRSfs_Ctx**(const *Npp32s* *pConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pConstants, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32s_C3IRSfs_Ctx**(const *Npp32s* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32s_C3IRSfs_Ctx**(const *Npp32s* *pConstants, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32sc_C1RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* nConstant, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32sc_C1IRSfs_Ctx**(const *Npp32sc* nConstant, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **nConstant** – host memory constant.

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32sc_C3RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* aConstants[3], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32sc_C3IRSfs_Ctx**(const *Npp32sc* aConstants[3], *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32sc_AC4RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* aConstants[3], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32sc_AC4IRSfs_Ctx**(const *Npp32sc* aConstants[3], *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* nConstant, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **nConstant** – host memory 32-bit floating point constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstant, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstant** – device memory 32-bit floating point constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16f_C1IR_Ctx**(const *Npp32f* nConstant, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image divided by constant.

Parameters

- ▶ **nConstant** – host memory 32-bit floating point constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16f_C1IR_Ctx**(const *Npp32f* *pConstant, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image divided by constant.

Parameters

- ▶ **pConstant** – device memory 32-bit floating point constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16f_C3R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16f_C3R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivC_16f_C3IR_Ctx(const Npp32f aConstants[3], Npp16f *pSrcDst, int  
                                nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
                                nppStreamCtx)
```

Three 16-bit floating point channel in place image divided by constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivDeviceC_16f_C3IR_Ctx(const Npp32f *pConstants, Npp16f *pSrcDst, int  
                                         nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
                                         nppStreamCtx)
```

Three 16-bit floating point channel in place image divided by constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivC_16f_C4R_Ctx(const Npp16f *pSrc1, int nSrc1Step, const Npp32f  
                                aConstants[4], Npp16f *pDst, int nDstStep, NppiSize oSizeROI,  
                                NppStreamContext nppStreamCtx)
```

Four 16-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16f_C4R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image divided by constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_16f_C4IR_Ctx**(const *Npp32f* aConstants[4], *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image divided by constant.

Parameters

- **aConstants** – fixed size host memory array of 32-bit floating point constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_16f_C4IR_Ctx**(const *Npp32f* *pConstants, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image divided by constant.

Parameters

- **pConstants** – fixed size device memory array of 32-bit floating point constant values, one per channel.

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* nConstant, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstant, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32f_C1IR_Ctx**(const *Npp32f* nConstant, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image divided by constant.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32f_C1IR_Ctx**(const *Npp32f* *pConstant, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image divided by constant.

Parameters

- ▶ **pConstant** – device memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivC_32f_C3IR_Ctx(const Npp32f aConstants[3], Npp32f *pSrcDst, int  
                                nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
                                nppStreamCtx)
```

Three 32-bit floating point channel in place image divided by constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDivDeviceC_32f_C3IR_Ctx(const Npp32f *pConstants, Npp32f *pSrcDst, int  
                                         nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
                                         nppStreamCtx)
```

Three 32-bit floating point channel in place image divided by constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[3], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32f_AC4IR_Ctx**(const *Npp32f* aConstants[3], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image divided by constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32f_AC4IR_Ctx**(const *Npp32f* *pConstants, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image divided by constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* aConstants[4], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pConstants, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32f_C4IR_Ctx**(const *Npp32f* aConstants[4], *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image divided by constant.

Parameters

- ▶ **aConstants** – fixed size host memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivDeviceC_32f_C4IR_Ctx**(const *Npp32f* *pConstants, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image divided by constant.

Parameters

- ▶ **pConstants** – fixed size device memory array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32fc_C1R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* nConstant, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32fc_C1IR_Ctx**(const *Npp32fc* nConstant, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

Parameters

- ▶ **nConstant** – host memory constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32fc_C3R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[3], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32fc_C3IR_Ctx**(const *Npp32fc* aConstants[3], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32fc_AC4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[3], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image divided by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32fc_AC4IR_Ctx**(const *Npp32fc* aConstants[3], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image divided by constant.

Parameters

- ▶ **aConstants** – fixed size array of host memory constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32fc_C4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* aConstants[4], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDivC_32fc_C4IR_Ctx**(const *Npp32fc* aConstants[4], *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

Parameters

- **aConstants** – fixed size array of host memory constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.7 AbsDiffC

Determines absolute difference between each pixel of an image and a constant value.

Note: If you use one of the device constant versions of these functions and the function called immediately preceeding that function generates that device constant you MUST either call `cudaStreamSynchronize()` or `cudaDeviceSynchronize()` before calling the device constant function.

Functions

NppStatus **nppiAbsDiffC_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nConstant, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image absolute difference with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbsDiffDeviceC_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* *pConstant, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image absolute difference with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAbsDiffC_16u_C1R_Ctx(const Npp16u *pSrc1, int nSrc1Step, Npp16u *pDst, int  
nDstStep, NppiSize oSizeROI, Npp16u nConstant,  
NppStreamContext nppStreamCtx)
```

One 16-bit unsigned short channel image absolute difference with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAbsDiffDeviceC_16u_C1R_Ctx(const Npp16u *pSrc1, int nSrc1Step, Npp16u  
*pDst, int nDstStep, NppiSize oSizeROI, Npp16u  
*pConstant, NppStreamContext nppStreamCtx)
```

One 16-bit unsigned short channel image absolute difference with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAbsDiffC_32f_C1R_Ctx(const Npp32f *pSrc1, int nSrc1Step, Npp32f *pDst, int  
nDstStep, NppiSize oSizeROI, Npp32f nConstant,  
NppStreamContext nppStreamCtx)
```

One 32-bit floating point channel image absolute difference with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – host memory constant.
- ▶ **pDst** – *Destination-Image Pointer*.

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbsDiffDeviceC_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* *pConstant, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image absolute difference with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pConstant** – device memory constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.8 Add

Pixel by pixel addition of two images.

Functions

NppStatus **nppiAdd_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image addition, scale by $2^l - nScaleFactor$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_8u_C1IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image addition, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image addition, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_8u_C3IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_8u_AC4IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_8u_C4IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image addition, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16u_C1IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image addition, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_16u_C3RSfs_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2,
                                int nSrc2Step, Npp16u *pDst, int nDstStep, NppiSize
                                oSizeROI, int nScaleFactor, NppStreamContext
                                nppStreamCtx)
```

Three 16-bit unsigned short channel image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_16u_C3IRSfs_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u *pSrcDst, int
                                nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,
                                NppStreamContext nppStreamCtx)
```

Three 16-bit unsigned short channel in place image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_16u_AC4RSfs_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2,
                                int nSrc2Step, Npp16u *pDst, int nDstStep, NppiSize
                                oSizeROI, int nScaleFactor, NppStreamContext
                                nppStreamCtx)
```

Four 16-bit unsigned short channel with unmodified alpha image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16u_AC4IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*

- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16u_C4IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image addition, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image addition, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_16s_C3IRSfs_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

Three 16-bit signed short channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_16s_AC4RSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s *pSrc2,  
int nSrc2Step, Npp16s *pDst, int nDstStep, NppiSize  
oSizeROI, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Four 16-bit signed short channel with unmodified alpha image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_16s_AC4IRSfs_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

Four 16-bit signed short channel with unmodified alpha in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image addition, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16s_C4IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image addition, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*

- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16sc_C3RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16sc_C3IRSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16sc_AC4RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image addition, scale by $2^{\text{int } nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16sc_AC4RSfs**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image addition, scale by $2^{\text{int } nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16sc_AC4IRSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image addition, scale by $2^{\text{int } nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*

- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image addition, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image add. Add the pixel values of corresponding pixels in the ROI and write them to the output image.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32s_C1IRSfs_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image addition, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32s_C3IRSfs_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32sc_C1RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32sc_C1IRSfs_Ctx**(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by $2^{\text{scaleFactor}}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_32sc_C3RSfs_Ctx(const Npp32sc *pSrc1, int nSrc1Step, const Npp32sc *pSrc2, int nSrc2Step, Npp32sc *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor, NppStreamContext nppStreamCtx)
```

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by $2^{\text{scaleFactor}}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_32sc_C3IRSfs_Ctx(const Npp32sc *pSrc, int nSrcStep, Npp32sc *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor, NppStreamContext nppStreamCtx)
```

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by $2^{\text{scaleFactor}}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32sc_AC4RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition, scale by $2^{\lceil -nScaleFactor \rceil}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32sc_AC4IRSfs_Ctx**(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition, scale by $2^{\lceil -nScaleFactor \rceil}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image addition.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16f_C1IR_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image addition.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16f_C3R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image addition.

Parameters

- **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16f_C3IR_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image addition.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_16f_C4R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image addition.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_16f_C4IR_Ctx(const Npp16f *pSrc, int nSrcStep, Npp16f *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
nppStreamCtx)
```

Four 16-bit floating point channel in place image addition.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_32f_C1R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f *pSrc2, int  
nSrc2Step, Npp32f *pDst, int nDstStep, NppiSize oSizeROI,  
NppStreamContext nppStreamCtx)
```

One 32-bit floating point channel image addition.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAdd_32f_C1IR_Ctx(const Npp32f *pSrc, int nSrcStep, Npp32f *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
nppStreamCtx)
```

One 32-bit floating point channel in place image addition.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image addition.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32f_C3IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image addition.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha image addition.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32f_AC4IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image addition.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image addition.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiAdd_32f_C4IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image addition.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiAdd_32fc_C1R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiAdd_32fc_C1IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32fc_C3R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32fc_C3IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32fc_AC4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32fc_AC4IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32fc_C4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAdd_32fc_C4IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.9 AddSquare

Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.

Functions

NppStatus **nppiAddSquare_8u32f_C1IMR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pMask** – *Mask-Image Pointer.*
- ▶ **nMaskStep** – *Mask-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*

- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddSquare_8u32f_C1IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image squared then added to in place floating point destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddSquare_16u32f_C1IMR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pMask** – *Mask-Image Pointer.*
- ▶ **nMaskStep** – *Mask-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddSquare_16u32f_C1IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image squared then added to in place floating point destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddSquare_32f_C1IMR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pMask** – *Mask-Image Pointer.*
- ▶ **nMaskStep** – *Mask-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddSquare_32f_C1IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image squared then added to in place floating point destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*

- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.10 AddProduct

Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.

Functions

NppStatus **nppiAddProduct_8u32f_C1IMR_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pMask** – *Mask-Image Pointer.*
- ▶ **nMaskStep** – *Mask-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddProduct_8u32f_C1IR_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image product added to in place floating point destination image.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddProduct_16u32f_C1IMR_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pMask** – *Mask-Image Pointer.*
- ▶ **nMaskStep** – *Mask-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddProduct_16u32f_C1IR_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image product added to in place floating point destination image.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*

- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddProduct_32f_C1IMR_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pMask** – *Mask-Image Pointer.*
- ▶ **nMaskStep** – *Mask-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddProduct_32f_C1IR_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image product added to in place floating point destination image.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddProduct_16f_C1IR_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image product added to in place floating point destination image.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.11 AddWeighted

Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.

Functions

NppStatus **nppiAddWeighted_8u32f_C1IMR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *Npp32f* nAlpha, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pMask** – *Mask-Image Pointer.*
- ▶ **nMaskStep** – *Mask-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAlpha** – Alpha weight to be applied to source image pixels (0.0F to 1.0F)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddWeighted_8u32f_C1IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *Npp32f* nAlpha, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAlpha** – Alpha weight to be applied to source image pixels (0.0F to 1.0F)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddWeighted_16u32f_C1IMR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *Npp32f* nAlpha, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pMask** – *Mask-Image Pointer.*
- ▶ **nMaskStep** – *Mask-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAlpha** – Alpha weight to be applied to source image pixels (0.0F to 1.0F)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddWeighted_16u32f_C1IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *Npp32f* nAlpha, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAlpha** – Alpha weight to be applied to source image pixels (0.0F to 1.0F)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddWeighted_32f_C1IMR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *Npp32f* nAlpha, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pMask** – *Mask-Image Pointer.*
- ▶ **nMaskStep** – *Mask-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAlpha** – Alpha weight to be applied to source image pixels (0.0F to 1.0F)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAddWeighted_32f_C1IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *Npp32f* nAlpha, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel alpha weighted image added to in place floating point destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAlpha** – Alpha weight to be applied to source image pixels (0.0F to 1.0F)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.12 Mul

Pixel by pixel multiply of two images.

Functions

NppStatus **nppiMul_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image multiplication, scale by $2^{\text{nScaleFactor}}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMul_8u_C1IRSfs_Ctx(const Npp8u *pSrc, int nSrcStep, Npp8u *pSrcDst, int  
                                nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
                                NppStreamContext nppStreamCtx)
```

One 8-bit unsigned char channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMul_8u_C3RSfs_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int  
                                nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, int  
                                nScaleFactor, NppStreamContext nppStreamCtx)
```

Three 8-bit unsigned char channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMul_8u_C3IRSfs_Ctx(const Npp8u *pSrc, int nSrcStep, Npp8u *pSrcDst, int  
                                nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
                                NppStreamContext nppStreamCtx)
```

Three 8-bit unsigned char channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image multiplication, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_8u_AC4IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image multiplication, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image multiplication, scale by $2^{\ell - nScaleFactor}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_8u_C4IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image multiplication, scale by $2^{\ell - nScaleFactor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16u_C1IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16u_C3IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image multiplication, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image multiplication, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16u_AC4IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image multiplication, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMul_16u_C4IRSfs_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

Four 16-bit unsigned short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMul_16s_C1IRSfs_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s *pSrc2,  
int nSrc2Step, Npp16s *pDst, int nDstStep, NppiSize  
oSizeROI, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

One 16-bit signed short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMul_16s_C1IRSfs_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,  
NppStreamContext nppStreamCtx)
```

One 16-bit signed short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16s_C3IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*

- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16s_AC4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha image multiplication, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16s_AC4IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha in place image multiplication, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16s_C4IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16sc_C3RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*

- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16sc_C3IRSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16sc_AC4RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16sc_AC4IRSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

1 channel 32-bit image multiplication. Multiply corresponding pixels in ROI.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32s_C1IRSfs_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32s_C3IRSfs_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image multiplication, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32sc_C1RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*

- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32sc_C1IRSfs_Ctx**(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32sc_C3RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32sc_C3IRSfs_Ctx**(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32sc_AC4RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32sc_AC4IRSfs_Ctx**(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image multiplication.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16f_C1IR_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image multiplication.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*

- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16f_C3R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image multiplication.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16f_C3IR_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image multiplication.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16f_C4R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image multiplication.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_16f_C4IR_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image multiplication.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image multiplication.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32f_C1IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image multiplication.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image multiplication.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32f_C3IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image multiplication.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha image multiplication.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32f_AC4IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image multiplication.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image multiplication.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32f_C4IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image multiplication.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32fc_C1R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32fc_C1IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32fc_C3R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32fc_C3IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32fc_AC4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32fc_AC4IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32fc_C4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMul_32fc_C4IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.13 MulScale

Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.

Functions

NppStatus **nppiMulScale_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_8u_C1IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_8u_C3IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_8u_AC4IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_8u_C4IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_16u_C1IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulScale_16u_C3R_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u
                                     *pSrc2, int nSrc2Step, Npp16u *pDst, int nDstStep,
                                     NppiSize oSizeROI, NppStreamContext nppStreamCtx)
```

Three 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulScale_16u_C3IR_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u *pSrcDst, int
                                      nSrcDstStep, NppiSize oSizeROI, NppStreamContext
                                      nppStreamCtx)
```

Three 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiMulScale_16u_AC4R_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u
                                     *pSrc2, int nSrc2Step, Npp16u *pDst, int nDstStep,
                                     NppiSize oSizeROI, NppStreamContext nppStreamCtx)
```

Four 16-bit unsigned short channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*

- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_16u_AC4IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMulScale_16u_C4IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.14 Sub

Pixel by pixel subtraction of two images.

Functions

NppStatus **nppiSub_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image subtraction, scale by $2^{\text{nScaleFactor}}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_8u_C1IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image subtraction, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image subtraction, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_8u_C3IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image subtraction, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image subtraction, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_8u_AC4IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image subtraction, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image subtraction, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_8u_C4IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image subtraction, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16u_C1IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16u_C3IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16u_AC4IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16u_C4IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16s_C1IRSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16s_C1IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16s_C3IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*

- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16s_AC4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha image subtraction, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16s_AC4IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha in place image subtraction, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image subtraction, scale by $2^{\ell - nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16s_C4IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image subtraction, scale by $2^{\ell - nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by $2^{\text{scaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by $2^{\text{scaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16sc_C3RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by $2^{\text{scaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16sc_C3IRSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16sc_AC4RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtraction, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16sc_AC4IRSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtraction, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image subtraction, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image subtraction. Subtract pSrc1's pixels from corresponding pixels in pSrc2.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32s_C1IRSfs_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32s_C3IRSfs_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32s_C4RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image subtraction, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*

- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32s_C4IRSfs_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image subtraction, scale by $2^{\text{scaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32sc_C1RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by $2^{\text{scaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32sc_C1IRSfs_Ctx**(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32sc_C3IRSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32sc_C3IRSfs_Ctx**(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32sc_AC4RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32sc_AC4IRSfs_Ctx**(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image subtraction.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16f_C1IR_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image subtraction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSub_16f_C3R_Ctx(const Npp16f *pSrc1, int nSrc1Step, const Npp16f *pSrc2, int  
nSrc2Step, Npp16f *pDst, int nDstStep, NppiSize oSizeROI,  
NppStreamContext nppStreamCtx)
```

Three 16-bit floating point channel image subtraction.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSub_16f_C3IR_Ctx(const Npp16f *pSrc, int nSrcStep, Npp16f *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
nppStreamCtx)
```

Three 16-bit floating point channel in place image subtraction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiSub_16f_C4R_Ctx(const Npp16f *pSrc1, int nSrc1Step, const Npp16f *pSrc2, int  
nSrc2Step, Npp16f *pDst, int nDstStep, NppiSize oSizeROI,  
NppStreamContext nppStreamCtx)
```

Four 16-bit floating point channel image subtraction.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_16f_C4IR_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image subtraction.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image subtraction.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32f_C1IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image subtraction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image subtraction.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32f_C3IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image subtraction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha image subtraction.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32f_AC4IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image subtraction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image subtraction.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32f_C4IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image subtraction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32fc_C1R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32fc_C1IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32fc_C3R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32fc_C3IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32fc_AC4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32fc_AC4IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32fc_C4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSub_32fc_C4IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.15 Div

Pixel by pixel division of two images.

Functions

NppStatus **nppiDiv_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image division, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_8u_C1IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image division, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image division, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_8u_C3IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image division, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha image division, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_8u_AC4IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel with unmodified alpha in place image division, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image division, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_8u_C4IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image division, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16u_C1RSfs**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*

- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16u_C1IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16u_C3IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image division, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha image division, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16u_AC4IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel with unmodified alpha in place image division, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16u_C4IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*

- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16s_C1IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16s_C3IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16s_AC4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha image division, scale by $2^{\text{C} - \text{nScaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16s_AC4IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel with unmodified alpha in place image division, scale by $2^{\text{C} - \text{nScaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image division, scale by $2^{\text{C} - \text{nScaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16s_C4IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16sc_C1RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16sc_C1IRSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16sc_C3RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16sc_C3IRSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16sc_AC4RSfs_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image division, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16sc_AC4IRSfs_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image division, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image division. Divide pixels in pSrc2 by pSrc1's pixels.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32s_C1IRSfs_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32s_C3RSfs_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32s_C3IRSfs_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image division, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32sc_C1RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32sc_C1IRSfs_Ctx**(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDiv_32sc_C3RSfs_Ctx(const Npp32sc *pSrc1, int nSrc1Step, const Npp32sc
                                   *pSrc2, int nSrc2Step, Npp32sc *pDst, int nDstStep,
                                   NppiSize oSizeROI, int nScaleFactor, NppStreamContext
                                   nppStreamCtx)
```

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDiv_32sc_C3IRSfs_Ctx(const Npp32sc *pSrc, int nSrcStep, Npp32sc *pSrcDst, int
                                   nSrcDstStep, NppiSize oSizeROI, int nScaleFactor,
                                   NppStreamContext nppStreamCtx)
```

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*

- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32sc_AC4RSfs_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division, scale by $2^{\text{c} - \text{nScaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32sc_AC4IRSfs_Ctx**(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division, scale by $2^{\text{c} - \text{nScaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image division.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16f_C1IR_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image division.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16f_C3R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image division.

Parameters

- **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16f_C3IR_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image division.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_16f_C4R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image division.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDiv_16f_C4IR_Ctx(const Npp16f *pSrc, int nSrcStep, Npp16f *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
nppStreamCtx)
```

Four 16-bit floating point channel in place image division.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDiv_32f_C1IR_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f *pSrc2, int  
nSrc2Step, Npp32f *pDst, int nDstStep, NppiSize oSizeROI,  
NppStreamContext nppStreamCtx)
```

One 32-bit floating point channel image division.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiDiv_32f_C1IR_Ctx(const Npp32f *pSrc, int nSrcStep, Npp32f *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
nppStreamCtx)
```

One 32-bit floating point channel in place image division.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image division.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32f_C3IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image division.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha image division.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32f_AC4IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel with unmodified alpha in place image division.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image division.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiDiv_32f_C4IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image division.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiDiv_32fc_C1R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiDiv_32fc_C1IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32fc_C3R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32fc_C3IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32fc_AC4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32fc_AC4IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32fc_C4R_Ctx**(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_32fc_C4IR_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.16 Div_Round

Pixel by pixel division of two images using result rounding modes.

Functions

NppStatus **nppiDiv_Round_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_8u_C3IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image division with unmodified alpha, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_8u_AC4IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image division with unmodified alpha, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image division, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_8u_C4IRSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16u_C1IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16u_C3IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image division, scale by $2^{\text{scaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image division with unmodified alpha, scale by $2^{\text{scaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16u_AC4IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image division with unmodified alpha, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image division, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16u_C4IRSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image division, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image division, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16s_C3IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image division, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16s_AC4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image division with unmodified alpha, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL)
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16s_AC4IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image division with unmodified alpha, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image division, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDiv_Round_16s_C4IRSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppRoundMode* rndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image division, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **rndMode** – Result Rounding mode to be used (NPP_RND_ZERO, NPP_RND_NEAR, or NP_RND_FINANCIAL).
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.17 Abs

Absolute value of each pixel value in an image.

Functions

NppStatus **nppiAbs_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image absolute value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16s_C1IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image absolute value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image absolute value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16s_C3IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image absolute value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image absolute value with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16s_AC4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image absolute value with unmodified alpha.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image absolute value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16s_C4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image absolute value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16f_C1R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image absolute value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16f_C1IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image absolute value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16f_C3R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image absolute value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16f_C3IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image absolute value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16f_C4R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image absolute value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_16f_C4IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image absolute value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image absolute value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_32f_C1IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image absolute value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image absolute value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image absolute value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image absolute value with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image absolute value with unmodified alpha.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image absolute value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbs_32f_C4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image absolute value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.18 AbsDiff

Pixel by pixel absolute difference between two images.

Functions

NppStatus **nppiAbsDiff_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel absolute difference of image1 minus image2.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAbsDiff_8u_C3R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int  
                                nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI,  
                                NppStreamContext nppStreamCtx)
```

Three 8-bit unsigned char channels absolute difference of image1 minus image2.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAbsDiff_8u_C4R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int  
                                nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI,  
                                NppStreamContext nppStreamCtx)
```

Four 8-bit unsigned char channels absolute difference of image1 minus image2.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAbsDiff_16u_C1R_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2,  
                                int nSrc2Step, Npp16u *pDst, int nDstStep, NppiSize  
                                oSizeROI, NppStreamContext nppStreamCtx)
```

One 16-bit unsigned short channel absolute difference of image1 minus image2.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*

- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbsDiff_16f_C1R_Ctx**(const *Npp16f* *pSrc1, int nSrc1Step, const *Npp16f* *pSrc2, int nSrc2Step, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel absolute difference of image1 minus image2.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAbsDiff_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel absolute difference of image1 minus image2.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.19 Sqr

Square each pixel in an image.

Functions

NppStatus **nppiSqr_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image squared, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_8u_C1IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image squared, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image squared, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_8u_C3IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image squared, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image squared with unmodified alpha, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_8u_AC4IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image squared with unmodified alpha, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image squared, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_8u_C4IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image squared, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image squared, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16u_C1IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image squared, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image squared, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16u_C3IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image squared, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image squared with unmodified alpha, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16u_AC4IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image squared with unmodified alpha, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16u_C4RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image squared, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16u_C4IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image squared, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image squared, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16s_C1IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image squared, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image squared, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*

- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16s_C3IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image squared, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16s_AC4RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image squared with unmodified alpha, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16s_AC4IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image squared with unmodified alpha, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16s_C4RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image squared, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16s_C4IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image squared, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16f_C1R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image squared.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16f_C1IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image squared.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16f_C3R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image squared.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16f_C3IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image squared.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*

- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16f_C4R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image squared.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_16f_C4IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image squared.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image squared.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_32f_C1IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image squared.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image squared.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image squared.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image squared with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image squared with unmodified alpha.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image squared.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqr_32f_C4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image squared.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.20 Sqrt

Pixel by pixel square root of each pixel in an image.

Functions

NppStatus **nppiSqrt_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image square root, scale by $2^{\ell - nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_8u_C1IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image square root, scale by $2^{\ell - nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image square root, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_8u_C3IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image square root, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_8u_AC4RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image square root with unmodified alpha, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*

- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_8u_AC4IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image square root with unmodified alpha, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image square root, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16u_C1IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image square root, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image square root, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16u_C3IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image square root, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16u_AC4RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image square root with unmodified alpha, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16u_AC4IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image square root with unmodified alpha, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image square root, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16s_C1IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image square root, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image square root, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16s_C3IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image square root, scale by $2^{\text{--}nScaleFactor}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*

- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16s_AC4RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image square root with unmodified alpha, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16s_AC4IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image square root with unmodified alpha, scale by $2^{\lfloor -nScaleFactor \rfloor}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16f_C1R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image square root.

Parameters

- **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16f_C1IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image square root.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16f_C3R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image square root.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16f_C3IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image square root.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16f_C4R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel image square root.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_16f_C4IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit floating point channel in place image square root.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image square root.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_32f_C1IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image square root.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image square root.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image square root.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image square root with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image square root with unmodified alpha.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image square root.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSqrt_32f_C4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel in place image square root.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.21 Ln

Pixel by pixel natural logarithm of each pixel in an image.

Functions

NppStatus **nppiLn_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_8u_C1IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image natural logarithm, scale by $2^{\text{--} nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_8u_C3IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image natural logarithm, scale by $2^{\text{--} nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image natural logarithm, scale by $2^{\text{--} nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16u_C1IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16u_C3IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*

- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16s_C1IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image natural logarithm, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16s_C3IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image natural logarithm, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16f_C1R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel image natural logarithm.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16f_C1IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit floating point channel in place image natural logarithm.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16f_C3R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel image natural logarithm.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_16f_C3IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit floating point channel in place image natural logarithm.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image natural logarithm.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_32f_C1IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image natural logarithm.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image natural logarithm.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLn_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image natural logarithm.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.1.22 Exp

Exponential value of each pixel in an image.

Functions

NppStatus **nppiExp_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image exponential, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_8u_C1IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image exponential, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image exponential, scale by $2^{(- nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_8u_C3IRSfs_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image exponential, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_16u_C1RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image exponential, scale by $2^{(nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_16u_C1IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_16u_C3RSfs_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_16u_C3IRSfs_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image exponential, scale by $2^{(-nScaleFactor)}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_16s_C1RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image exponential, scale by $2^{\ell - nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_16s_C1IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image exponential, scale by $2^{\ell - nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_16s_C3RSfs_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image exponential, scale by $2^{\ell - nScaleFactor}$, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*

- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_16s_C3IRSfs_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image exponential, scale by $2^{\text{nScaleFactor}}$, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image exponential.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_32f_C1IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel in place image exponential.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel image exponential.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiExp_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit floating point channel in place image exponential.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.2. Logical Operations

1.7.2.1 Logical Operations

The set of image processing logical operations available in the library.

1.7.2.2 AndC

Pixel by pixel logical and of an image with a constant.

Functions

NppStatus **nppiAndC_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image logical and with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_8u_C1IR_Ctx**(const *Npp8u* nConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image logical and with constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image logical and with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_8u_C3IR_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image logical and with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical and with constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_8u_AC4IR_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical and with constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical and with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_8u_C4IR_Ctx**(const *Npp8u* aConstants[4], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical and with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* nConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image logical and with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_16u_C1IR_Ctx**(const *Npp16u* nConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image logical and with constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image logical and with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_16u_C3IR_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image logical and with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image logical and with constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_16u_AC4IR_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image logical and with constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAndC_16u_C4R_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u
                                aConstants[4], Npp16u *pDst, int nDstStep, NppiSize oSizeROI,
                                NppStreamContext nppStreamCtx)
```

Four 16-bit unsigned short channel image logical and with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAndC_16u_C4IR_Ctx(const Npp16u aConstants[4], Npp16u *pSrcDst, int
                                nSrcDstStep, NppiSize oSizeROI, NppStreamContext
                                nppStreamCtx)
```

Four 16-bit unsigned short channel in place image logical and with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAndC_32s_C1R_Ctx(const Npp32s *pSrc1, int nSrc1Step, const Npp32s nConstant,
                                Npp32s *pDst, int nDstStep, NppiSize oSizeROI,
                                NppStreamContext nppStreamCtx)
```

One 32-bit signed integer channel image logical and with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_32s_C1IR_Ctx**(const *Npp32s* nConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image logical and with constant.

Parameters

- **nConstant** – Constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_32s_C3R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image logical and with constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_32s_C3IR_Ctx**(const *Npp32s* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image logical and with constant.

Parameters

- **aConstants** – fixed size array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_32s_AC4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical and with constant with unmodified alpha.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_32s_AC4IR_Ctx**(const *Npp32s* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical and with constant with unmodified alpha.

Parameters

- **aConstants** – fixed size array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_32s_C4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[4], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical and with constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of constant values, one per channel.

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAndC_32s_C4IR_Ctx**(const *Npp32s* aConstants[4], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical and with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.2.3 OrC

Pixel by pixel logical or of an image with a constant.

Functions

NppStatus **nppiOrC_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image logical or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_8u_C1IR_Ctx**(const *Npp8u* nConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image logical or with constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image logical or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_8u_C3IR_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image logical or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiOrC_8u_AC4R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u aConstants[3],
                             Npp8u *pDst, int nDstStep, NppiSize oSizeROI,
                             NppStreamContext nppStreamCtx)
```

Four 8-bit unsigned char channel image logical or with constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiOrC_8u_AC4IR_Ctx(const Npp8u aConstants[3], Npp8u *pSrcDst, int nSrcDstStep,
                               NppiSize oSizeROI, NppStreamContext nppStreamCtx)
```

Four 8-bit unsigned char channel in place image logical or with constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiOrC_8u_C4R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u aConstants[4],
                              Npp8u *pDst, int nDstStep, NppiSize oSizeROI, NppStreamContext
                              nppStreamCtx)
```

Four 8-bit unsigned char channel image logical or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_8u_C4IR_Ctx**(const *Npp8u* aConstants[4], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_16u_C1IR_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* nConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image logical or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_16u_C1IR_Ctx**(const *Npp16u* nConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image logical or with constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppi0rC_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image logical or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppi0rC_16u_C3IR_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image logical or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppi0rC_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image logical or with constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_16u_AC4IR_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image logical or with constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image logical or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_16u_C4IR_Ctx**(const *Npp16u* aConstants[4], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image logical or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* nConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image logical or with constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **nConstant** – Constant.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_32s_C1IR_Ctx**(const *Npp32s* nConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image logical or with constant.

Parameters

- **nConstant** – Constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_32s_C3R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image logical or with constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_32s_C3IR_Ctx**(const *Npp32s* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image logical or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_32s_AC4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical or with constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_32s_AC4IR_Ctx**(const *Npp32s* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical or with constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_32s_C4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[4], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOrC_32s_C4IR_Ctx**(const *Npp32s* aConstants[4], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.2.4 XorC

Pixel by pixel logical exclusive or of an image with a constant.

Functions

NppStatus **nppiXorC_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image logical exclusive or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_8u_C1IR_Ctx**(const *Npp8u* nConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image logical exclusive or with constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image logical exclusive or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_8u_C3IR_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image logical exclusive or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical exclusive or with constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_8u_AC4IR_Ctx**(const *Npp8u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical exclusive or with constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* aConstants[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical exclusive or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_8u_C4IR_Ctx**(const *Npp8u* aConstants[4], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical exclusive or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* nConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image logical exclusive or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_16u_C1IR_Ctx**(const *Npp16u* nConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image logical exclusive or with constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image logical exclusive or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_16u_C3IR_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image logical exclusive or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image logical exclusive or with constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_16u_AC4IR_Ctx**(const *Npp16u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image logical exclusive or with constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* aConstants[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image logical exclusive or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_16u_C4IR_Ctx**(const *Npp16u* aConstants[4], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image logical exclusive or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* nConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image logical exclusive or with constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_32s_C1IR_Ctx**(const *Npp32s* nConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image logical exclusive or with constant.

Parameters

- **nConstant** – Constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_32s_C3R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image logical exclusive or with constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_32s_C3IR_Ctx**(const *Npp32s* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image logical exclusive or with constant.

Parameters

- **aConstants** – fixed size array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_32s_AC4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical exclusive or with constant with unmodified alpha.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_32s_AC4IR_Ctx**(const *Npp32s* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical exclusive or with constant with unmodified alpha.

Parameters

- **aConstants** – fixed size array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_32s_C4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* aConstants[4], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical exclusive or with constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXorC_32s_C4IR_Ctx**(const *Npp32s* aConstants[4], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical exclusive or with constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.2.5 RShiftC

Pixel by pixel right shift of an image by a constant value.

Functions

NppStatus **nppiRShiftC_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp32u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8u_C1IR_Ctx**(const *Npp32u* nConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image right shift by constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8u_C3IR_Ctx**(const *Npp32u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image right shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image right shift by constant with unmodified alpha.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8u_AC4IR_Ctx**(const *Npp32u* aConstants[3], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image right shift by constant with unmodified alpha.

Parameters

- **aConstants** – fixed size array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image right shift by constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of constant values, one per channel.

- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8u_C4IR_Ctx**(const *Npp32u* aConstants[4], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image right shift by constant.

Parameters

- **aConstants** – fixed size array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8s_C1R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp32u* nConstant, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit signed char channel image right shift by constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **nConstant** – Constant.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8s_C1IR_Ctx**(const *Npp32u* nConstant, *Npp8s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit signed char channel in place image right shift by constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8s_C3R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit signed char channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8s_C3IR_Ctx**(const *Npp32u* aConstants[3], *Npp8s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit signed char channel in place image right shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8s_AC4R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit signed char channel image right shift by constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8s_AC4IR_Ctx**(const *Npp32u* aConstants[3], *Npp8s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit signed char channel in place image right shift by constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_8s_C4R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[4], *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit signed char channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiRShiftC_8s_C4IR_Ctx( const Npp32u aConstants[4], Npp8s *pSrcDst, int  
                                   nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
                                   nppStreamCtx)
```

Four 8-bit signed char channel in place image right shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiRShiftC_16u_C1R_Ctx( const Npp16u *pSrc1, int nSrc1Step, const Npp32u  
                                   nConstant, Npp16u *pDst, int nDstStep, NppiSize oSizeROI,  
                                   NppStreamContext nppStreamCtx)
```

One 16-bit unsigned short channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiRShiftC_16u_C1IR_Ctx( const Npp32u nConstant, Npp16u *pSrcDst, int  
                                   nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
                                   nppStreamCtx)
```

One 16-bit unsigned short channel in place image right shift by constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16u_C3IR_Ctx**(const *Npp32u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image right shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image right shift by constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16u_AC4IR_Ctx**(const *Npp32u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image right shift by constant with unmodified alpha.

Parameters

- **aConstants** – fixed size array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image right shift by constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16u_C4IR_Ctx**(const *Npp32u* aConstants[4], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image right shift by constant.

Parameters

- **aConstants** – fixed size array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16s_C1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp32u* nConstant, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16s_C1IR_Ctx**(const *Npp32u* nConstant, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel in place image right shift by constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16s_C3R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16s_C3IR_Ctx**(const *Npp32u* aConstants[3], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit signed short channel in place image right shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16s_AC4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image right shift by constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16s_AC4IR_Ctx**(const *Npp32u* aConstants[3], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image right shift by constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16s_C4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[4], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_16s_C4IR_Ctx**(const *Npp32u* aConstants[4], *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit signed short channel in place image right shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32u* nConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_32s_C1IR_Ctx**(const *Npp32u* nConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image right shift by constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_32s_C3R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_32s_C3IR_Ctx**(const *Npp32u* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image right shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_32s_AC4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image right shift by constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_32s_AC4IR_Ctx**(const *Npp32u* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image right shift by constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_32s_C4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[4], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image right shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiRShiftC_32s_C4IR_Ctx**(const *Npp32u* aConstants[4], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image right shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.2.6 LShiftC

Pixel by pixel left shift of an image by a constant value.

Functions

NppStatus **nppiLShiftC_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp32u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image left shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_8u_C1IR_Ctx**(const *Npp32u* nConstant, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image left shift by constant.

Parameters

- ▶ **nConstant** – Constant.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image left shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiLShiftC_8u_C3IR_Ctx( const Npp32u aConstants[3], Npp8u *pSrcDst, int  
                                   nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
                                   nppStreamCtx)
```

Three 8-bit unsigned char channel in place image left shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiLShiftC_8u_AC4R_Ctx( const Npp8u *pSrc1, int nSrc1Step, const Npp32u  
                                   aConstants[3], Npp8u *pDst, int nDstStep, NppiSize  
                                   oSizeROI, NppStreamContext nppStreamCtx)
```

Four 8-bit unsigned char channel image left shift by constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiLShiftC_8u_AC4IR_Ctx( const Npp32u aConstants[3], Npp8u *pSrcDst, int  
                                   nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
                                   nppStreamCtx)
```

Four 8-bit unsigned char channel in place image left shift by constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image left shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_8u_C4IR_Ctx**(const *Npp32u* aConstants[4], *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image left shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp32u* nConstant, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image left shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – Constant
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_16u_C1IR_Ctx**(const *Npp32u* nConstant, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image left shift by constant.

Parameters

- **nConstant** – Constant.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image left shift by constant.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **aConstants** – fixed size array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_16u_C3IR_Ctx**(const *Npp32u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image left shift by constant.

Parameters

- **aConstants** – fixed size array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image left shift by constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_16u_AC4IR_Ctx**(const *Npp32u* aConstants[3], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image left shift by constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image left shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_16u_C4IR_Ctx**(const *Npp32u* aConstants[4], *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image left shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32u* nConstant, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image left shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nConstant** – Constant.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_32s_C1IR_Ctx**(const *Npp32u* nConstant, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image left shift by constant.

Parameters

- **nConstant** – Constant.
- **pSrcDst** – *In-Place Image Pointer*.
- **nSrcDstStep** – *In-Place-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_32s_C3R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image left shift by constant.

Parameters

- **pSrc1** – *Source-Image Pointer*.
- **nSrc1Step** – *Source-Image Line Step*.
- **aConstants** – fixed size array of constant values, one per channel.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_32s_C3IR_Ctx**(const *Npp32u* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image left shift by constant.

Parameters

- **aConstants** – fixed size array of constant values, one per channel.
- **pSrcDst** – *In-Place Image Pointer*.
- **nSrcDstStep** – *In-Place-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_32s_AC4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image left shift by constant with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_32s_AC4IR_Ctx**(const *Npp32u* aConstants[3], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image left shift by constant with unmodified alpha.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_32s_C4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32u* aConstants[4], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image left shift by constant.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiLShiftC_32s_C4IR_Ctx**(const *Npp32u* aConstants[4], *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image left shift by constant.

Parameters

- ▶ **aConstants** – fixed size array of constant values, one per channel.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.2.7 And

Pixel by pixel logical and of images.

Functions

NppStatus **nppiAnd_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image logical and.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_8u_C1IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image logical and.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image logical and.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_8u_C3IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image logical and.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical and with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_8u_AC4IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical and with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical and.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_8u_C4IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical and.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image logical and.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_16u_C1IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image logical and.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image logical and.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_16u_C3IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image logical and.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAnd_16u_AC4R_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int  
nSrc2Step, Npp16u *pDst, int nDstStep, NppiSize oSizeROI,  
NppStreamContext nppStreamCtx)
```

Four 16-bit unsigned short channel image logical and with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAnd_16u_AC4IR_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
nppStreamCtx)
```

Four 16-bit unsigned short channel in place image logical and with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAnd_16u_C4R_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int  
nSrc2Step, Npp16u *pDst, int nDstStep, NppiSize oSizeROI,  
NppStreamContext nppStreamCtx)
```

Four 16-bit unsigned short channel image logical and.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_16u_C4IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image logical and.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image logical and.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_32s_C1IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image logical and.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_32s_C3R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image logical and.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_32s_C3IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image logical and.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_32s_AC4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical and with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_32s_AC4IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical and with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_32s_C4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical and.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAnd_32s_C4IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical and.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.2.8 Or

Pixel by pixel logical or of images.

Functions

NppStatus **nppiOr_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image logical or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_8u_C1IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image logical or.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image logical or.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_8u_C3IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image logical or.

Parameters

- **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical or with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_8u_AC4IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical or with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_8u_C4IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image logical or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_16u_C1IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image logical or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image logical or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_16u_C3IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image logical or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image logical or with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_16u_AC4IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image logical or with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiOr_16u_C4R_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int  
nSrc2Step, Npp16u *pDst, int nDstStep, NppiSize oSizeROI,  
NppStreamContext nppStreamCtx)
```

Four 16-bit unsigned short channel image logical or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiOr_16u_C4IR_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u *pSrcDst, int  
nSrcDstStep, NppiSize oSizeROI, NppStreamContext  
nppStreamCtx)
```

Four 16-bit unsigned short channel in place image logical or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiOr_32s_C1R_Ctx(const Npp32s *pSrc1, int nSrc1Step, const Npp32s *pSrc2, int  
nSrc2Step, Npp32s *pDst, int nDstStep, NppiSize oSizeROI,  
NppStreamContext nppStreamCtx)
```

One 32-bit signed integer channel image logical or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_32s_C1IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image logical or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_32s_C3R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image logical or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_32s_C3IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image logical or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_32s_AC4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical or with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_32s_AC4IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical or with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_32s_C4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiOr_32s_C4IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.2.9 Xor

Pixel by pixel logical exclusive or of images.

Functions

NppStatus **nppiXor_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image logical exclusive or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_8u_C1IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image logical exclusive or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image logical exclusive or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.

- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_8u_C3IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image logical exclusive or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical exclusive or with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_8u_AC4IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical exclusive or with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical exclusive or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_8u_C4IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical exclusive or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*

- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image logical exclusive or.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_16u_C1IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel in place image logical exclusive or.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image logical exclusive or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_16u_C3IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image logical exclusive or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image logical exclusive or with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiXor_16u_AC4IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image logical exclusive or with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiXor_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image logical exclusive or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiXor_16u_C4IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image logical exclusive or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image logical exclusive or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_32s_C1IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel in place image logical exclusive or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_32s_C3R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel image logical exclusive or.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_32s_C3IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 32-bit signed integer channel in place image logical exclusive or.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_32s_AC4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical exclusive or with unmodified alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_32s_AC4IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical exclusive or with unmodified alpha.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_32s_C4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image logical exclusive or.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiXor_32s_C4IR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel in place image logical exclusive or.

Parameters

- **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.2.10 Not

Pixel by pixel logical not of image.

Functions

NppStatus **nppiNot_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image logical not.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiNot_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image logical not.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiNot_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image logical not.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiNot_8u_C3IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image logical not.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiNot_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical not with unmodified alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiNot_8u_AC4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical not with unmodified alpha.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiNot_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image logical not.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiNot_8u_C4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image logical not.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.3. Image Alpha Composition Operations

1.7.3.1 AlphaCompC

Composite two images using constant alpha values.

Functions

NppStatus **nppiAlphaCompC_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* nAlpha1, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* nAlpha2, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image composition using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaCompC_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* nAlpha1, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* nAlpha2, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image composition using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer*.

- **nSrc2Step** – *Source-Image Line Step.*
- **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **eAlphaOp** – alpha-blending operation.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaCompC_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* nAlpha1, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* nAlpha2, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image composition using constant alpha.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **eAlphaOp** – alpha-blending operation.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaCompC_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* nAlpha1, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* nAlpha2, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image composition with alpha using constant source alpha.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*

- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaCompC_8s_C1R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, *Npp8s* nAlpha1, const *Npp8s* *pSrc2, int nSrc2Step, *Npp8s* nAlpha2, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 8-bit signed char channel image composition using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaCompC_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, *Npp16u* nAlpha1, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* nAlpha2, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image composition using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaCompC_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, *Npp16u* nAlpha1, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* nAlpha2, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel image composition using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaCompC_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, *Npp16u* nAlpha1, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* nAlpha2, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image composition using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaCompC_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, *Npp16u* nAlpha1, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* nAlpha2, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image composition with alpha using constant source alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaCompC_16s_C1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, *Npp16s* nAlpha1, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* nAlpha2, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image composition using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaCompC_32u_C1R_Ctx**(const *Npp32u* *pSrc1, int nSrc1Step, *Npp32u* nAlpha1, const *Npp32u* *pSrc2, int nSrc2Step, *Npp32u* nAlpha2, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 32-bit unsigned integer channel image composition using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAlphaCompC_32s_C1R_Ctx( const Npp32s *pSrc1, int nSrc1Step, Npp32s nAlpha1,
                                         const Npp32s *pSrc2, int nSrc2Step, Npp32s nAlpha2,
                                         Npp32s *pDst, int nDstStep, NppiSize oSizeROI,
                                         NppiAlphaOp eAlphaOp, NppStreamContext
                                         nppStreamCtx)
```

One 32-bit signed integer channel image composition using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **nAlpha2** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAlphaCompC_32f_C1R_Ctx( const Npp32f *pSrc1, int nSrc1Step, Npp32f nAlpha1,
                                         const Npp32f *pSrc2, int nSrc2Step, Npp32f nAlpha2,
                                         Npp32f *pDst, int nDstStep, NppiSize oSizeROI,
                                         NppiAlphaOp eAlphaOp, NppStreamContext
                                         nppStreamCtx)
```

One 32-bit floating point channel image composition using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0.0 - 1.0).
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **nAlpha2** – Image alpha opacity (0.0 - 1.0).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

Premultiplies pixels of an image using a constant alpha value.

Functions

NppStatus **nppiAlphaPremulC_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* nAlpha1, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image premultiplication using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiAlphaPremulC_8u_C1IR_Ctx**(*Npp8u* nAlpha1, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel in place image premultiplication using constant alpha.

Parameters

- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiAlphaPremulC_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* nAlpha1, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel image premultiplication using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremulC_8u_C3IR_Ctx**(*Npp8u* nAlpha1, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 8-bit unsigned char channel in place image premultiplication using constant alpha.

Parameters

- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremulC_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* nAlpha1, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image premultiplication using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremulC_8u_C4IR_Ctx**(*Npp8u* nAlpha1, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image premultiplication using constant alpha.

Parameters

- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremulC_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* nAlpha1, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image premultiplication with alpha using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremulC_8u_AC4IR_Ctx**(*Npp8u* nAlpha1, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image premultiplication with alpha using constant alpha.

Parameters

- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAlphaPremulC_16u_C1R_Ctx( const Npp16u *pSrc1, int nSrc1Step, Npp16u
                                         nAlpha1, Npp16u *pDst, int nDstStep, NppiSize
                                         oSizeROI, NppStreamContext nppStreamCtx )
```

One 16-bit unsigned short channel image premultiplication using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAlphaPremulC_16u_C1IR_Ctx( Npp16u nAlpha1, Npp16u *pSrcDst, int
                                         nSrcDstStep, NppiSize oSizeROI,
                                         NppStreamContext nppStreamCtx )
```

One 16-bit unsigned short channel in place image premultiplication using constant alpha.

Parameters

- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiAlphaPremulC_16u_C3R_Ctx( const Npp16u *pSrc1, int nSrc1Step, Npp16u
                                         nAlpha1, Npp16u *pDst, int nDstStep, NppiSize
                                         oSizeROI, NppStreamContext nppStreamCtx )
```

Three 16-bit unsigned short channel image premultiplication using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremulC_16u_C3IR_Ctx**(*Npp16u* nAlpha1, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three 16-bit unsigned short channel in place image premultiplication using constant alpha.

Parameters

- **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremulC_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, *Npp16u* nAlpha1, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image premultiplication using constant alpha.

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremulC_16u_C4IR_Ctx**(*Npp16u* nAlpha1, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image premultiplication using constant alpha.

Parameters

- **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremulC_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, *Npp16u* nAlpha1, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image premultiplication with alpha using constant alpha.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremulC_16u_AC4IR_Ctx**(*Npp16u* nAlpha1, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image premultiplication with alpha using constant alpha.

Parameters

- ▶ **nAlpha1** – Image alpha opacity (0 - max channel pixel value).
- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.7.3.2 AlphaComp

Composite two images using alpha opacity values contained in each image.

Functions

NppStatus **nppiAlphaComp_8u_AC1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eAlphaOp** – alpha-blending operation.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_8s_AC1R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 8-bit signed char channel image composition using image alpha values (0 - max channel pixel value).

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **eAlphaOp** – alpha-blending operation.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_16u_AC1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pSrc2** – *Source-Image Pointer.*
- **nSrc2Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **eAlphaOp** – alpha-blending operation.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_16s_AC1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 16-bit signed short channel image composition using image alpha values (0 - max channel pixel value).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_32u_AC1R_Ctx**(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_32u_AC4R_Ctx**(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Four 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eAlphaOp** – alpha-blending operation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_32s_AC1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eAlphaOp** – *alpha-blending operation.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_32s_AC4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Four 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eAlphaOp** – *alpha-blending operation.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_32f_AC1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

One 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*

- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eAlphaOp** – *alpha-blending operation.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaComp_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiAlphaOp* eAlphaOp, *NppStreamContext* nppStreamCtx)

Four 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eAlphaOp** – *alpha-blending operation.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

Premultiplies image pixels by image alpha opacity values.

Functions

NppStatus **nppiAlphaPremu1_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel image premultiplication with pixel alpha (0 - max channel pixel value).

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*

- **nSrc1Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremul_8u_AC4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 8-bit unsigned char channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremul_16u_AC4IR_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel image premultiplication with pixel alpha (0 - max channel pixel value).

Parameters

- **pSrc1** – *Source-Image Pointer.*
- **nSrc1Step** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiAlphaPremul_16u_AC4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four 16-bit unsigned short channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

Parameters

- **pSrcDst** – *In-Place Image Pointer.*

- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.8. Image Color Conversion Functions

Routines manipulating an image's color model and sampling format.

These functions can be found in the nppicc library. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

1.8.1. Color Processing Functions

Routines for performing image color manipulation.

RGBToYUVColorTwist

Normally, color twist is a 3-channel operation that takes place on RGB data.

The following functions also perform the color twist operation, but while converting between image formats: RGBToYUV420, RGBToYUV422, and RGBToNV12.

To recap color twist pixel processing: Color twist consists of applying the following formula to each image pixel using coefficients from the user supplied color twist host matrix array as follows where $dst[x]$ and $src[x]$ represent destination pixel and source pixel channel or plane x . The full sized coefficient matrix should be sent for all pixel channel sizes, the function will process the appropriate coefficients and channels for the corresponding pixel size.

This is how the matrix works for a RGB->YUV420/YUV422/NV12 forward transform:

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪ aTwist[0][3]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪ aTwist[1][3]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪ aTwist[2][3]
```

```
NppStatus nppiRGBToYUV420_8u_ColorTwist32f_P3R_Ctx(const Npp8u *const pSrc[3], int
                                                    aSrcStep[3], Npp8u *pDst[3], int
                                                    aDstStep[3], NppiSize oSizeROI,
                                                    const Npp32f aTwist[3][4],
                                                    NppStreamContext nppStreamCtx)
```

Three channel 8-bit unsigned planar RGB conversion to three channel 8-bit unsigned planar YUV 4:2:0, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYUV420_16u_ColorTwist32f_P3R_Ctx**(const *Npp16u* *const pSrc[3], int aSrcStep[3], *Npp16u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned planar RGB conversion to three channel 16-bit unsigned planar YUV 4:2:0, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYUV420_8u_ColorTwist32f_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned packed RGB conversion to three channel 8-bit unsigned planar YUV 4:2:0, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYUV420_16u_ColorTwist32f_C3P3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned packed RGB conversion to three channel 16-bit unsigned planar YUV 4:2:0, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYUV422_8u_ColorTwist32f_P3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned planar RGB conversion to three channel 8-bit unsigned planar YUV 4:2:2, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiRGBToYUV422_16u_ColorTwist32f_P3R_Ctx**(const *Npp16u* *const pSrc[3], int aSrcStep[3], *Npp16u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned planar RGB conversion to three channel 16-bit unsigned planar YUV 4:2:2, using a Color Twist to compute the exact color space arithmetic.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **aSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **aDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiRGBToYUV422_8u_ColorTwist32f_C3C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned packed RGB conversion to two channel 8-bit unsigned packed YUV 4:2:2, using a Color Twist to compute the exact color space arithmetic.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYUV422_16u_ColorTwist32f_C3C2R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned packed RGB conversion to two channel 16-bit unsigned packed YUV 4:2:2, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYUV422_8u_ColorTwist32f_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned packed RGB conversion to three channel 8-bit unsigned planar YUV 4:2:2, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiRGBToYUV422_16u_ColorTwist32f_C3P3R_Ctx(const Npp16u *pSrc, int
                                                         nSrcStep, Npp16u *pDst[3], int
                                                         aDstStep[3], NppiSize oSizeROI,
                                                         const Npp32f aTwist[3][4],
                                                         NppStreamContext
                                                         nppStreamCtx)
```

Three channel 16-bit unsigned packed RGB conversion to three channel 16-bit unsigned planar YUV 4:2:2, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiRGBToNV12_8u_ColorTwist32f_C3P2R_Ctx(const Npp8u *pSrc, int nSrcStep,
                                                         Npp8u *pDst[2], int aDstStep[2],
                                                         NppiSize oSizeROI, const Npp32f
                                                         aTwist[3][4], NppStreamContext
                                                         nppStreamCtx)
```

Three channel 8-bit unsigned packed RGB conversion to two channel 8-bit unsigned planar NV12, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiRGBToNV12_16u_ColorTwist32f_C3P2R_Ctx(const Npp16u *pSrc, int nSrcStep,
                                                         Npp16u *pDst[2], int aDstStep[2],
                                                         NppiSize oSizeROI, const Npp32f
                                                         aTwist[3][4], NppStreamContext
                                                         nppStreamCtx)
```

Three channel 16-bit unsigned packed RGB conversion to two channel 16-bit unsigned planar NV12, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiRGBToNV12_8u_ColorTwist32f_P3P2R_Ctx(const Npp8u *const pSrc[3], int  
aSrcStep[3], Npp8u *pDst[2], int  
aDstStep[2], NppiSize oSizeROI,  
const Npp32f aTwist[3][4],  
NppStreamContext nppStreamCtx)
```

Three channel 8-bit unsigned planar RGB conversion to two channel 8-bit unsigned planar NV12, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiRGBToNV12_16u_ColorTwist32f_P3P2R_Ctx(const Npp16u *const pSrc[3], int  
aSrcStep[3], Npp16u *pDst[2], int  
aDstStep[2], NppiSize oSizeROI,  
const Npp32f aTwist[3][4],  
NppStreamContext nppStreamCtx)
```

Three channel 16-bit unsigned planar RGB conversion to two channel 16-bit unsigned planar NV12, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

YUVToRGBColorTwist

This is the inverse color twist, a 3-channel operation that takes place on YUV/NV12 data to produce RGB data, supporting the following conversions: YUV420ToRGB, YUV422ToRGB, and NV12ToRGB.

The INVERSE Color twist consists of applying the following formula to each image pixel using coefficients from the user supplied color twist host matrix array as follows where `dst[x]` and `src[x]` represent destination pixel and source pixel channel or plane x. The full sized coefficient matrix should be sent for all pixel channel sizes, the function will process the appropriate coefficients and channels for the corresponding pixel size.

This is how the matrix works for the YUV420/YUV/422/NV12->RGB INVERSE transform:

(note- do the offsets first):

```
src[0]' = src[0] + aTwist[0][3]
src[1]' = src[1] + aTwist[1][3]
src[2]' = src[2] + aTwist[2][3]
```

And then the remaining 3x3 twist matrix is applied using those modified values:

```
dst[0] = aTwist[0][0] * src[0]' + aTwist[0][1] * src[1]' + aTwist[0][2] * src[2]'
dst[1] = aTwist[1][0] * src[0]' + aTwist[1][1] * src[1]' + aTwist[1][2] * src[2]'
dst[2] = aTwist[2][0] * src[0]' + aTwist[2][1] * src[1]' + aTwist[2][2] * src[2]'
```

Since the 4th column of the matrix is the offsets (either 0 or half-max to shift the values to be positive), as they're applied last in the forward transform (to YUV), they need to be applied FIRST in the inverse transform (back to RGB). This does mean that +-16384 has to be used for 16u images where there was a +-128 for 8u images in both forward and inverse cases.

```
NppStatus nppiYUV420ToRGB_8u_ColorTwist32f_P3R_Ctx(const Npp8u *const pSrc[3], int
aSrcStep[3], Npp8u *pDst[3], int
aDstStep[3], NppiSize oSizeROI,
const Npp32f aTwist[3][4],
NppStreamContext nppStreamCtx)
```

Three channel 8-bit unsigned planar YUV4:2:0 conversion to three channel 8-bit unsigned planar RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV420ToRGB_16u_ColorTwist32f_P3R_Ctx**(const *Npp16u* *const pSrc[3], int aSrcStep[3], *Npp16u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned planar YUV4:2:0 conversion to three channel 16-bit unsigned planar RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV420ToRGB_8u_ColorTwist32f_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned planar YUV4:2:0 conversion to three channel 8-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*

- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV420ToRGB_16u_ColorTwist32f_P3C3R_Ctx**(const *Npp16u* *const pSrc[3], int aSrcStep[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned planar YUV4:2:0 conversion to three channel 16-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV420ToRGB_8u_ColorTwist32f_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned planar YUV4:2:0 conversion to four channel 8-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic, with constant alpha (0xFF).

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV420ToRGB_16u_ColorTwist32f_P3C4R_Ctx**(const *Npp16u* *const pSrc[3], int aSrcStep[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned planar YUV4:2:0 conversion to four channel 16-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic, with constant alpha (0xFFFF).

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV420ToRGB_8u_ColorTwist32f_P3AC4R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], const *Npp8u* nAlpha, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned planar YUV4:2:0 conversion to four channel 8-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic, with user set alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nAlpha** – 8-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV420ToRGB_16u_ColorTwist32f_P3AC4R_Ctx**(const *Npp16u* *const pSrc[3],
int aSrcStep[3], *Npp16u* *pDst,
int nDstStep, *NppiSize*
oSizeROI, const *Npp32f*
aTwist[3][4], const *Npp16u*
nAlpha, *NppStreamContext*
nppStreamCtx)

Three channel 16-bit unsigned planar YUV4:2:0 conversion to four channel 16-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic, with user set alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nAlpha** – 16-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV422ToRGB_8u_ColorTwist32f_C2C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep,
Npp8u *pDst, int nDstStep,
NppiSize oSizeROI, const *Npp32f*
aTwist[3][4], *NppStreamContext*
nppStreamCtx)

Two channel 8-bit unsigned packed YUV4:2:2 conversion to three channel 8-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV422ToRGB_16u_ColorTwist32f_C2C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Two channel 16-bit unsigned packed YUV4:2:2 conversion to three channel 16-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV422ToRGB_8u_ColorTwist32f_P3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned planar YUV4:2:2 conversion to three channel 8-bit unsigned planar RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV422ToRGB_16u_ColorTwist32f_P3R_Ctx**(const *Npp16u* *const pSrc[3], int aSrcStep[3], *Npp16u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned planar YUV4:2:2 conversion to three channel 16-bit unsigned planar RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **aSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **aDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV422ToRGB_8u_ColorTwist32f_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned planar YUV4:2:2 conversion to three channel 8-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **aSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiYUV422ToRGB_16u_ColorTwist32f_P3C3R_Ctx(const Npp16u *const pSrc[3], int  
aSrcStep[3], Npp16u *pDst, int  
nDstStep, NppiSize oSizeROI,  
const Npp32f aTwist[3][4],  
NppStreamContext  
nppStreamCtx)
```

Three channel 8-bit unsigned planar YUV4:2:2 conversion to three channel 8-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiYUV422ToRGB_8u_ColorTwist32f_P3AC4R_Ctx(const Npp8u *const pSrc[3], int  
aSrcStep[3], Npp8u *pDst, int  
nDstStep, NppiSize oSizeROI,  
const Npp32f aTwist[3][4],  
const Npp8u nAlpha,  
NppStreamContext  
nppStreamCtx)
```

Three channel 8-bit unsigned planar YUV4:2:2 conversion to three channel 8-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic, with user set alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nAlpha** – 8-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiYUV422ToRGB_16u_ColorTwist32f_P3AC4R_Ctx(const Npp16u *const pSrc[3],
                                                         int aSrcStep[3], Npp16u *pDst,
                                                         int nDstStep, NppiSize
                                                         oSizeROI, const Npp32f
                                                         aTwist[3][4], const Npp16u
                                                         nAlpha, NppStreamContext
                                                         nppStreamCtx)
```

Three channel 16-bit unsigned planar YUV4:2:2 conversion to three channel 16-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic, with user set alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nAlpha** – 16-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiNV12ToRGB_8u_ColorTwist32f_P2C3R_Ctx(const Npp8u *const pSrc[2], int
                                                         aSrcStep[2], Npp8u *pDst, int
                                                         nDstStep, NppiSize oSizeROI, const
                                                         Npp32f aTwist[3][4],
                                                         NppStreamContext nppStreamCtx)
```

Two channel 8-bit unsigned planar NV12 conversion to three channel 8-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiNV12ToRGB_16u_ColorTwist32f_P2C3R_Ctx(const Npp16u *const pSrc[2], int  
aSrcStep[2], Npp16u *pDst, int  
nDstStep, NppiSize oSizeROI, const  
Npp32f aTwist[3][4],  
NppStreamContext nppStreamCtx)
```

Two channel 16-bit unsigned planar NV12 conversion to three channel 16-bit unsigned packed RGB, using a Color Twist to compute the exact color space arithmetic.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

RGBToUYVP Conversion

Three unsigned 8-bit packed RGB channels color space conversion to three channel unsigned packed 10-bit UYVP pixel.

param pSrc
Source-Image Pointer.

param aSrcStep
Source-Image Line Step.

param oSrcOffset
pSrc pixel offset relative to image origin, currently must be 0,0.

param pDst
Destination-Image Pointer.

param aDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param eColorSpace
The input color space.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes.


```

NppStatus nppiRGB_8u_ToUYVP_10u_C3C3R_Ctx( const Npp8u *pSrc, int nSrcStep, NppiPoint
                                             oSrcOffset, Npp8u *pDst, int nDstStep, NppiSize
                                             oSizeROI, NppiColorSpace eColorSpace,
                                             NppStreamContext nppStreamCtx)

NppStatus nppiRGB_8u_ToUYVP_10u_P3C3R_Ctx( const Npp8u *const pSrc[3], const int nSrcStep,
                                             NppiPoint oSrcOffset, Npp8u *pDst, int
                                             nDstStep, NppiSize oSizeROI, NppiColorSpace
                                             eColorSpace, NppStreamContext
                                             nppStreamCtx)

NppStatus nppiRGB_8u_ToUYVP_10u_C3P3R_Ctx( const Npp8u *const pSrc, const int nSrcStep,
                                             NppiPoint oSrcOffset, Npp8u *pDst[3], int
                                             nDstStep, NppiSize oSizeROI, NppiColorSpace
                                             eColorSpace, NppStreamContext
                                             nppStreamCtx)

NppStatus nppiRGBA_8u_ToUYVP_10u_AC4C3R_Ctx( const Npp8u *const pSrc, const int nSrcStep,
                                                NppiPoint oSrcOffset, Npp8u *pDst, int
                                                nDstStep, NppiSize oSizeROI, NppiColorSpace
                                                eColorSpace, NppStreamContext
                                                nppStreamCtx)

NppStatus nppiRGB_16u_ToUYVP_10u_C3C3R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiPoint
                                             oSrcOffset, Npp8u *pDst, int nDstStep,
                                             NppiSize oSizeROI, NppiColorSpace
                                             eColorSpace, NppStreamContext
                                             nppStreamCtx)

NppStatus nppiRGB_16u_ToUYVP_10u_P3C3R_Ctx( const Npp16u *const pSrc[3], const int
                                             nSrcStep, NppiPoint oSrcOffset, Npp8u *pDst,
                                             int nDstStep, NppiSize oSizeROI,
                                             NppiColorSpace eColorSpace,
                                             NppStreamContext nppStreamCtx)

NppStatus nppiRGBA_16u_ToUYVP_10u_AC4C3R_Ctx( const Npp16u *const pSrc, const int
                                                nSrcStep, NppiPoint oSrcOffset, Npp8u
                                                *pDst, int nDstStep, NppiSize oSizeROI,
                                                NppiColorSpace eColorSpace,
                                                NppStreamContext nppStreamCtx)

NppStatus nppiRGB_16f_ToUYVP_10u_C3C3R_Ctx( const Npp16f *pSrc, int nSrcStep, NppiPoint
                                             oSrcOffset, Npp8u *pDst, int nDstStep,
                                             NppiSize oSizeROI, NppiColorSpace
                                             eColorSpace, NppStreamContext
                                             nppStreamCtx)

NppStatus nppiRGB_16f_ToUYVP_10u_P3C3R_Ctx( const Npp16f *const pSrc[3], const int
                                             nSrcStep, NppiPoint oSrcOffset, Npp8u *pDst,
                                             int nDstStep, NppiSize oSizeROI,
                                             NppiColorSpace eColorSpace,
                                             NppStreamContext nppStreamCtx)

```

```
NppStatus nppiRGBA_16f_ToUYVP_10u_AC4C3R_Ctx(const Npp16f *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp8u *pDst, int nDstStep,
NppiSize oSizeROI, NppiColorSpace
eColorSpace, NppStreamContext
nppStreamCtx)
```

UYVPToRGB Conversion

Three channel unsigned packed 10-bit UYVP pixel color space conversion to three unsigned packed 8-bit channels RGB.

param pSrc

Source-Image Pointer.

param aSrcStep

Source-Image Line Step.

param oSrcOffset

pSrc pixel offset relative to image origin, currently must be 0,0.

param pDst

Destination-Image Pointer.

param aDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eColorSpace

The output color space.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes.

```
NppStatus nppiUYVP_10u_ToRGB_8u_C3C3R_Ctx(const Npp8u *pSrc, int srcStep, NppiPoint
oSrcOffset, Npp8u *pDst, int dstStep, NppiSize
oSizeROI, NppiColorSpace eColorSpace,
NppStreamContext nppStreamCtx)
```

```
NppStatus nppiUYVP_10u_ToRGB_8u_C3AC4R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp8u *pDst, int nDstStep,
NppiSize oSizeROI, NppiColorSpace
eColorSpace, Npp8u alpha, NppStreamContext
nppStreamCtx)
```

```
NppStatus nppiUYVP_10u_ToRGB_8u_C3P3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp8u *pDst[3], const int nDstStep,
NppiSize oSizeROI, NppiColorSpace eColorSpace,
NppStreamContext nppStreamCtx)
```

```

NppStatus nppiUYVP_10u_ToRGB_16u_C3C3R_Ctx(const Npp8u *pSrc, int srcStep, NppiPoint
oSrcOffset, Npp16u *pDst, int dstStep,
NppiSize oSizeROI, NppiColorSpace
eColorSpace, NppStreamContext
nppStreamCtx)

NppStatus nppiUYVP_10u_ToRGB_16u_C3AC4R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp16u *pDst, int nDstStep,
NppiSize oSizeROI, NppiColorSpace
eColorSpace, Npp16u alpha,
NppStreamContext nppStreamCtx)

NppStatus nppiUYVP_10u_ToRGB_16u_C3P3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp16u *pDst[3], const int
nDstStep, NppiSize oSizeROI, NppiColorSpace
eColorSpace, NppStreamContext
nppStreamCtx)

NppStatus nppiUYVP_10u_ToRGB_16f_C3C3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp16f *pDst, int nDstStep,
NppiSize oSizeROI, NppiColorSpace
eColorSpace, NppStreamContext
nppStreamCtx)

NppStatus nppiUYVP_10u_ToRGBA_16f_C3AC4R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp16f *pDst, int nDstStep,
NppiSize oSizeROI, NppiColorSpace
eColorSpace, Npp32f nAlphaValue,
NppStreamContext nppStreamCtx)

NppStatus nppiUYVP_10u_ToRGB_16f_C3P3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp16f *const pDst[3], const int
nDstStep, NppiSize oSizeROI, NppiColorSpace
eColorSpace, NppStreamContext
nppStreamCtx)

```

UYVPTranscoding

UYVP 10-bit packed YCbCr 4:2:2 transcoding to/from NV12, P010, P208, NV16, V210, P210, I444, YUV444, Y410.

```

NppStatus nppiNV12_8u_ToUYVP_10u_P2C3R_Ctx(const Npp8u *const pSrc[2], const int
nSrcStep[2], NppiPoint oSrcOffset, Npp8u
*pDst, int nDstStep, NppiSize oSizeROI,
NppStreamContext nppStreamCtx)

```

4:2:0 conversions

```

NppStatus nppiP010_16u_ToUYVP_10u_P3C3R_Ctx(const Npp16u *const pSrc[3], const int
nSrcStep, NppiPoint oSrcOffset, Npp8u *pDst,
int nDstStep, NppiSize oSizeROI,
NppStreamContext nppStreamCtx)

```

```

NppStatus nppiUYVP_10u_ToP208_8u_C3C3R_Ctx(const Npp8u *pSrc, int srcStep, NppiPoint
oSrcOffset, Npp8u *pDst, int dstStep, NppiSize
oSizeROI, NppStreamContext nppStreamCtx)

```

4:2:2 conversions

```
NppStatus nppiUYVP_10u_ToNV16_8u_C3P2R_Ctx(const Npp8u *const pSrc, const int nSrcStep,
                                             NppiPoint oSrcOffset, Npp8u *pDst[2], const
                                             int nDstStep[2], NppiSize oSizeROI,
                                             NppStreamContext nppStreamCtx)
```

```
NppStatus nppiUYVP_10u_ToV210_10u_C3C3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
                                             oSrcOffset, Npp8u *pDst, int nDstStep,
                                             NppiSize oSizeROI, NppStreamContext
                                             nppStreamCtx)
```

```
NppStatus nppiP208_8u_ToUYVP_10u_C3C3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
                                             oSrcOffset, Npp8u *pDst, int nDstStep,
                                             NppiSize oSizeROI, NppStreamContext
                                             nppStreamCtx)
```

```
NppStatus nppiNV16_8u_ToUYVP_10u_P2C3R_Ctx(const Npp8u *const pSrc[2], const int
                                             nSrcStep[2], NppiPoint oSrcOffset, Npp8u
                                             *pDst, int nDstStep, NppiSize oSizeROI,
                                             NppStreamContext nppStreamCtx)
```

```
NppStatus nppiV210_10u_ToUYVP_10u_C3C3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
                                             oSrcOffset, Npp8u *pDst, int nDstStep,
                                             NppiSize oSizeROI, NppStreamContext
                                             nppStreamCtx)
```

```
NppStatus nppiP210_16u_ToUYVP_10u_P2C3R_Ctx(const Npp16u *const pSrc[2], const int
                                             nSrcStep[2], NppiPoint oSrcOffset, Npp8u
                                             *pDst, int nDstStep, NppiSize oSizeROI,
                                             NppStreamContext nppStreamCtx)
```

```
NppStatus nppiUYVP_10u_ToI444_8u_C3P3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
                                             oSrcOffset, Npp8u *const pDst[3], const int
                                             nDstStep, NppiSize oSizeROI,
                                             NppStreamContext nppStreamCtx)
```

4:4:4 conversions

```
NppStatus nppiUYVP_10u_ToYUV444_8u_C3C3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
                                             oSrcOffset, Npp8u *pDst, int nDstStep,
                                             NppiSize oSizeROI, NppStreamContext
                                             nppStreamCtx)
```

```
NppStatus nppiUYVP_10u_ToY410_10u_C3C3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiPoint
                                             oSrcOffset, Npp8u *pDst, int nDstStep,
                                             NppiSize oSizeROI, NppStreamContext
                                             nppStreamCtx)
```

```
NppStatus nppiI444_8u_ToUYVP_10u_P3C3R_Ctx(const Npp8u *const pSrc[3], const int
                                             nSrcStep, NppiPoint oSrcOffset, Npp8u *pDst,
                                             int nDstStep, NppiSize oSizeROI,
                                             NppStreamContext nppStreamCtx)
```

```
NppStatus nppiYUV444_8u_ToUYVP_10u_C3C3R_Ctx( const Npp8u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp8u *pDst, int nDstStep,
NppiSize oSizeROI, NppStreamContext
nppStreamCtx )
```

```
NppStatus nppiY410_10u_ToUYVP_10u_C3C3R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp8u *pDst, int nDstStep,
NppiSize oSizeROI, NppStreamContext
nppStreamCtx )
```

RGBToUYVY Conversion

```
NppStatus nppiRGB_8u_ToUYVY_8u_C3C3R_Ctx( const Npp8u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp8u *pDst, int nDstStep, NppiSize
oSizeROI, NppiColorSpace eColorSpace,
NppStreamContext nppStreamCtx )
```

```
NppStatus nppiRGB_8u_ToUYVY_8u_P3C3R_Ctx( const Npp8u *const pSrc[3], const int nSrcStep,
NppiPoint oSrcOffset, Npp8u *pDst, int nDstStep,
NppiSize oSizeROI, NppiColorSpace eColorSpace,
NppStreamContext nppStreamCtx )
```

```
NppStatus nppiRGBA_8u_ToUYVY_8u_AC4C3R_Ctx( const Npp8u *const pSrc, const int nSrcStep,
NppiPoint oSrcOffset, Npp8u *pDst, int
nDstStep, NppiSize oSizeROI, NppiColorSpace
eColorSpace, NppStreamContext
nppStreamCtx )
```

```
NppStatus nppiRGB_16u_ToUYVY16_16u_C3C3R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiPoint
oSrcOffset, Npp16u *pDst, int nDstStep,
NppiSize oSizeROI, NppiColorSpace
eColorSpace, NppStreamContext
nppStreamCtx )
```

```
NppStatus nppiRGB_16u_ToUYVY16_16u_P3C3R_Ctx( const Npp16u *const pSrc[3], const int
nSrcStep, NppiPoint oSrcOffset, Npp16u
*pDst, int nDstStep, NppiSize oSizeROI,
NppiColorSpace eColorSpace,
NppStreamContext nppStreamCtx )
```

```
NppStatus nppiRGBA_16u_ToUYVY16_16u_AC4C3R_Ctx( const Npp16u *const pSrc, const int
nSrcStep, NppiPoint oSrcOffset, Npp16u
*pDst, int nDstStep, NppiSize oSizeROI,
NppiColorSpace eColorSpace,
NppStreamContext nppStreamCtx )
```

UYVYToRGB Conversion

NppStatus **nppiUYVY_8u_ToRGB_8u_C3C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiColorSpace* eColorSpace, *NppStreamContext* nppStreamCtx)

NppStatus **nppiUYVY_8u_ToRGB_8u_C3AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiColorSpace* eColorSpace, *Npp8u* nAlpha, *NppStreamContext* nppStreamCtx)

NppStatus **nppiUYVY_8u_ToRGB_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiPoint* oSrcOffset, *Npp8u* *pDst[3], const int nDstStep, *NppiSize* oSizeROI, *NppiColorSpace* eColorSpace, *NppStreamContext* nppStreamCtx)

NppStatus **nppiUYVY16_16u_ToRGB_16u_C3C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiColorSpace* eColorSpace, *NppStreamContext* nppStreamCtx)

NppStatus **nppiUYVY16_16u_ToRGB_16u_C3AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiColorSpace* eColorSpace, *Npp16u* nAlpha, *NppStreamContext* nppStreamCtx)

NppStatus **nppiUYVY16_16u_ToRGB_16u_C3P3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiPoint* oSrcOffset, *Npp16u* *pDst[3], const int nDstStep, *NppiSize* oSizeROI, *NppiColorSpace* eColorSpace, *NppStreamContext* nppStreamCtx)

NV12ToRGBExt NV12 to RGB/BGR Extended Conversions

NppStatus **nppiNV12ToRGB_8u_P2C4R_Ctx**(const *Npp8u* *const pSrc[2], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiColorSpace* eColorSpace, *Npp8u* nAlpha, *NppStreamContext* nppStreamCtx)

NppStatus **nppiNV12ToBGR_8u_P2C4R_Ctx**(const *Npp8u* *const pSrc[2], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiColorSpace* eColorSpace, *Npp8u* nAlpha, *NppStreamContext* nppStreamCtx)

NppStatus **nppiNV12ToRGB_BT2020_8u_P2C3R_Ctx**(const *Npp8u* *const pSrc[2], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

```
NppStatus nppiNV12ToBGR_BT2020_8u_P2C3R_Ctx(const Npp8u *const pSrc[2], int nSrcStep,
                                              Npp8u *pDst, int nDstStep, NppiSize oSizeROI,
                                              NppStreamContext nppStreamCtx)
```

P010ToRGB P010 to RGB/RGBA Conversions

```
NppStatus nppiP010ToRGB_8u_P2C3R_Ctx(const Npp16u *const pSrc[2], int nSrcStep, Npp8u
                                       *pDst, int nDstStep, NppiSize oSizeROI,
                                       NppiColorSpace eColorSpace, NppStreamContext
                                       nppStreamCtx)
```

```
NppStatus nppiP010ToRGB_8u_P2C4R_Ctx(const Npp16u *const pSrc[2], int nSrcStep, Npp8u
                                       *pDst, int nDstStep, NppiSize oSizeROI,
                                       NppiColorSpace eColorSpace, Npp8u nAlpha,
                                       NppStreamContext nppStreamCtx)
```

```
NppStatus nppiP010ToRGB_16u_P2C3R_Ctx(const Npp16u *const pSrc[2], int nSrcStep, Npp16u
                                       *pDst, int nDstStep, NppiSize oSizeROI,
                                       NppiColorSpace eColorSpace, NppStreamContext
                                       nppStreamCtx)
```

```
NppStatus nppiP010ToRGB_16u_P2C4R_Ctx(const Npp16u *const pSrc[2], int nSrcStep, Npp16u
                                       *pDst, int nDstStep, NppiSize oSizeROI,
                                       NppiColorSpace eColorSpace, Npp16u nAlpha,
                                       NppStreamContext nppStreamCtx)
```

P016ToRGB P016 to RGB/RGBA Conversions

```
NppStatus nppiP016ToRGB_16u_P2C3R_Ctx(const Npp16u *const pSrc[2], int nSrcStep, Npp16u
                                       *pDst, int nDstStep, NppiSize oSizeROI,
                                       NppiColorSpace eColorSpace, NppStreamContext
                                       nppStreamCtx)
```

```
NppStatus nppiP016ToRGB_16u_P2C4R_Ctx(const Npp16u *const pSrc[2], int nSrcStep, Npp16u
                                       *pDst, int nDstStep, NppiSize oSizeROI,
                                       NppiColorSpace eColorSpace, Npp16u nAlpha,
                                       NppStreamContext nppStreamCtx)
```

RGBToNV12 RGB to NV12 Conversions

```
NppStatus nppiRGBToNV12_8u_C3P2R_Ctx(const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[2], int
                                       nDstStep, NppiSize oSizeROI, NppiColorSpace
                                       eColorSpace, NppStreamContext nppStreamCtx)
```

```
NppStatus nppiRGBToNV12_8u_P3P2R_Ctx(const Npp8u *const pSrc[3], int nSrcStep, Npp8u
                                       *pDst[2], int nDstStep, NppiSize oSizeROI,
                                       NppiColorSpace eColorSpace, NppStreamContext
                                       nppStreamCtx)
```


YUV420ToNV12 YUV420 to NV12 Format Repack

```
NppStatus nppiYUV420ToNV12_8u_P3P2R_Ctx(const Npp8u *const pSrc[3], const int aSrcStep[3],  
                                           Npp8u *pDst[2], int nDstStep, NppiSize oSizeROI,  
                                           NppStreamContext nppStreamCtx)
```

1.8.1.1 Color To Gray Conversion

Routines for converting color images to grayscale.

RGBToGray

RGB to CCIR601 Gray conversion.

Here is how NPP converts gamma corrected RGB to CCIR601 Gray.

```
nGray = 0.299F * R + 0.587F * G + 0.114F * B;
```

```
NppStatus nppiRGBToGray_8u_C3C1R_Ctx(const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int  
                                       nDstStep, NppiSize oSizeROI, NppStreamContext  
                                       nppStreamCtx)
```

3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiRGBToGray_8u_AC4C1R_Ctx(const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int  
                                       nDstStep, NppiSize oSizeROI, NppStreamContext  
                                       nppStreamCtx)
```

4 channel 8-bit unsigned packed RGB with alpha to 1 channel 8-bit unsigned packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToGray_16u_C3C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToGray_16u_AC4C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned packed RGB with alpha to 1 channel 16-bit unsigned packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToGray_16s_C3C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToGray_16s_AC4C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed packed RGB with alpha to 1 channel 16-bit signed packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToGray_32f_C3C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToGray_32f_AC4C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point packed RGB with alpha to 1 channel 32-bit floating point packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

ColorToGray

RGB Color to Gray conversion using user supplied conversion coefficients.

Here is how NPP converts gamma corrected RGB Color to Gray using user supplied conversion coefficients.

```
nGray = aCoeffs[0] * R + aCoeffs[1] * G + aCoeffs[2] * B;
```

For the C4C1R versions of the functions the calculations are as follows.

For BGRA or other formats with alpha just rearrange the coefficients accordingly.

```
nGray = aCoeffs[0] * R + aCoeffs[1] * G + aCoeffs[2] * B + aCoeffs[3] * A;
```

NppStatus **nppiColorToGray_8u_C3C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aCoeffs[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorToGray_8u_AC4C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aCoeffs[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 1 channel 8-bit unsigned packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorToGray_8u_C4C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aCoeffs[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGBA to 1 channel 8-bit unsigned packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorToGray_16u_C3C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aCoeffs[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.

- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorToGray_16u_AC4C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aCoeffs[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned packed RGB with alpha to 1 channel 16-bit unsigned packed Gray conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorToGray_16u_C4C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aCoeffs[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned packed RGBA to 1 channel 16-bit unsigned packed Gray conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiColorToGray_16s_C3C1R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst,  
int nDstStep, NppiSize oSizeROI, const Npp32f  
aCoeffs[3], NppStreamContext nppStreamCtx)
```

3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiColorToGray_16s_AC4C1R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst,  
int nDstStep, NppiSize oSizeROI, const Npp32f  
aCoeffs[3], NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed packed RGB with alpha to 1 channel 16-bit signed packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiColorToGray_16s_C4C1R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst,  
int nDstStep, NppiSize oSizeROI, const Npp32f  
aCoeffs[4], NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed packed RGBA to 1 channel 16-bit signed packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorToGray_32f_C3C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aCoeffs[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorToGray_32f_AC4C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aCoeffs[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point packed RGB with alpha to 1 channel 32-bit floating point packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorToGray_32f_C4C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aCoeffs[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point packed RGBA to 1 channel 32-bit floating point packed Gray conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aCoeffs** – Host memory fixed size array of constant floating point conversion coefficient values, one per color channel.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

GradientColorToGray

RGB Color to Gray Gradient conversion using user selected gradient distance method.

NppStatus **nppiGradientColorToGray_8u_C3C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiNorm* eNorm, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray Gradient conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eNorm** – Gradient distance method to use.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiGradientColorToGray_16u_C3C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiNorm* eNorm, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray Gradient conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eNorm** – Gradient distance method to use.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiGradientColorToGray_16s_C3C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiNorm* eNorm, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray Gradient conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eNorm** – Gradient distance method to use.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiGradientColorToGray_32f_C3C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiNorm* eNorm, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray Gradient conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eNorm** – Gradient distance method to use.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.1.2 Color Debayer

Grayscale Color Filter Array to RGB Color Debayer conversion.

Generates one RGB color pixel for every grayscale source pixel. Source and destination images must have even width and height. Missing pixel colors are generated using bilinear interpolation with chroma correlation of generated green values (eInterpolation MUST be set to 0). eGrid allows the user to specify the Bayer grid registration position at source image location oSrcROI.x, oSrcROI.y relative to pSrc. Possible registration positions are:

NPPI_BAYER_BGGR	NPPI_BAYER_RGGG	NPPI_BAYER_GBRG	NPPI_BAYER_GRBG
B G G R	R G G B	G B R G	G R B G

If it becomes necessary to access source pixels outside source image then the source image borders are mirrored.

Here is how the algorithm works. R, G, and B base pixels from the source image are used unmodified. To generate R values for those G pixels, the average of $R(x - 1, y)$ and $R(x + 1, y)$ or $R(x, y - 1)$ and $R(x, y + 1)$ is used depending on whether the left and right or top and bottom pixels are R base pixels. To generate B values for those G pixels, the same algorithm is used using nearest B values. For an R base pixel, if there are no B values in the upper, lower, left, or right adjacent pixels then B is the average of B values in the 4 diagonal (G base) pixels. The same algorithm is used using R values to generate the R value of a B base pixel. Chroma correlation is applied to generated G values only, for a B base pixel $G(x - 1, y)$ and $G(x + 1, y)$ are averaged or $G(x, y - 1)$ and $G(x, y + 1)$ are averaged depending on whether the absolute difference between $B(x, y)$ and the average of $B(x - 2, y)$ and $B(x + 2, y)$ is smaller than the absolute difference between $B(x, y)$ and the average of $B(x, y - 2)$ and $B(x, y + 2)$. For an R base pixel the same algorithm is used testing against the surrounding R values at those offsets. If the horizontal and vertical differences are the same at one of those pixels then the average of the four left, right, upper and lower G values is used instead.

Functions

NppStatus **nppiCFAToRGB_8u_C1C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiBayerGridPosition* eGrid, *NppiInterpolationMode* eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned packed CFA grayscale Bayer pattern to 3 channel 8-bit unsigned packed RGB conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSrcSize** – full source image width and height relative to pSrc.
- ▶ **oSrcROI** – rectangle specifying starting source image pixel x and y location relative to pSrc and ROI width and height.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **eGrid** – enumeration value specifying bayer grid registration position at location oSrcROI.x, oSrcROI.y relative to pSrc.
- ▶ **eInterpolation** – MUST be NPPI_INTER_UNDEFINED
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCFAToRGBA_8u_C1AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiBayerGridPosition* eGrid, *NppiInterpolationMode* eInterpolation, *Npp8u* nAlpha, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned packed CFA grayscale Bayer pattern to 4 channel 8-bit unsigned packed RGB conversion with alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSrcSize** – full source image width and height relative to pSrc.
- ▶ **oSrcROI** – rectangle specifying starting source image pixel x and y location relative to pSrc and ROI width and height.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **eGrid** – enumeration value specifying bayer grid registration position at location oSrcROI.x, oSrcROI.y relative to pSrc.
- ▶ **eInterpolation** – MUST be NPPI_INTER_UNDEFINED

- ▶ **nAlpha** – constant alpha value to be written to each destination pixel
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCFAToRGB_16u_C1C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiRect* oSrcROI, *Npp16u* *pDst, int nDstStep, *NppiBayerGridPosition* eGrid, *NppiInterpolationMode* eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned packed CFA grayscale Bayer pattern to 3 channel 16-bit unsigned packed RGB conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSrcSize** – full source image width and height relative to pSrc.
- ▶ **oSrcROI** – rectangle specifying starting source image pixel x and y location relative to pSrc and ROI width and height.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **eGrid** – enumeration value specifying bayer grid registration position at location oSrcROI.x, oSrcROI.y relative to pSrc.
- ▶ **eInterpolation** – MUST be NPPI_INTER_UNDEFINED
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCFAToRGBA_16u_C1AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiRect* oSrcROI, *Npp16u* *pDst, int nDstStep, *NppiBayerGridPosition* eGrid, *NppiInterpolationMode* eInterpolation, *Npp16u* nAlpha, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned packed CFA grayscale Bayer pattern to 4 channel 16-bit unsigned packed RGB conversion with alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSrcSize** – full source image width and height relative to pSrc.
- ▶ **oSrcROI** – rectangle specifying starting source image pixel x and y location relative to pSrc and ROI width and height.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.

- ▶ **eGrid** – enumeration value specifying bayer grid registration position at location oSrcROI.x, oSrcROI.y relative to pSrc.
- ▶ **eInterpolation** – MUST be NPPI_INTER_UNDEFINED
- ▶ **nAlpha** – constant alpha value to be written to each destination pixel
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCFAToRGB_32u_C1C3R_Ctx**(const *Npp32u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiRect* oSrcROI, *Npp32u* *pDst, int nDstStep, *NppiBayerGridPosition* eGrid, *NppiInterpolationMode* eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 32-bit unsigned packed CFA grayscale Bayer pattern to 3 channel 32-bit unsigned packed RGB conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSrcSize** – full source image width and height relative to pSrc.
- ▶ **oSrcROI** – rectangle specifying starting source image pixel x and y location relative to pSrc and ROI width and height.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **eGrid** – enumeration value specifying bayer grid registration position at location oSrcROI.x, oSrcROI.y relative to pSrc.
- ▶ **eInterpolation** – MUST be NPPI_INTER_UNDEFINED
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCFAToRGBA_32u_C1AC4R_Ctx**(const *Npp32u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiRect* oSrcROI, *Npp32u* *pDst, int nDstStep, *NppiBayerGridPosition* eGrid, *NppiInterpolationMode* eInterpolation, *Npp32u* nAlpha, *NppStreamContext* nppStreamCtx)

1 channel 32-bit unsigned packed CFA grayscale Bayer pattern to 4 channel 32-bit unsigned packed RGB conversion with alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSrcSize** – full source image width and height relative to pSrc.
- ▶ **oSrcROI** – rectangle specifying starting source image pixel x and y location relative to pSrc and ROI width and height.

- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **eGrid** – enumeration value specifying bayer grid registration position at location oSrcROI.x, oSrcROI.y relative to pSrc.
- ▶ **eInterpolation** – MUST be NPPI_INTER_UNDEFINED
- ▶ **nAlpha** – constant alpha value to be written to each destination pixel
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.1.3 Color Gamma Correction

Routines for correcting image color gamma.

GammaFwd

Forward gamma correction.

NppStatus **nppiGammaFwd_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed color not in place forward gamma correction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiGammaFwd_8u_C3IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed color in place forward gamma correction.

Parameters

- ▶ **pSrcDst** – in place packed pixel image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiGammaFwd_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed color with alpha not in place forward gamma correction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiGammaFwd_8u_AC4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed color with alpha in place forward gamma correction.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiGammaFwd_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar color not in place forward gamma correction.

Parameters

- ▶ **pSrc** – source planar pixel format image pointer array.
- ▶ **nSrcStep** – source planar pixel format image line step.
- ▶ **pDst** – destination planar pixel format image pointer array.
- ▶ **nDstStep** – destination planar pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiGammaFwd_8u_IP3R_Ctx**(*Npp8u* *const pSrcDst[3], int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar color in place forward gamma correction.

Parameters

- ▶ **pSrcDst** – in place planar pixel format image pointer array.
- ▶ **nSrcDstStep** – in place planar pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

GammaInv

Inverse gamma correction.

NppStatus **nppiGammaInv_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed color not in place inverse gamma correction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiGammaInv_8u_C3IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed color in place inverse gamma correction.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiGammaInv_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed color with alpha not in place inverse gamma correction.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiGammaInv_8u_AC4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed color with alpha in place inverse gamma correction.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiGammaInv_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar color not in place inverse gamma correction.

Parameters

- ▶ **pSrc** – source planar pixel format image pointer array.
- ▶ **nSrcStep** – source planar pixel format image line step.
- ▶ **pDst** – destination planar pixel format image pointer array.
- ▶ **nDstStep** – destination planar pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiGammaInv_8u_IP3R_Ctx**(*Npp8u* *const pSrcDst[3], int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar color in place inverse gamma correction.

Parameters

- ▶ **pSrcDst** – in place planar pixel format image pointer array.
- ▶ **nSrcDstStep** – in place planar pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.1.4 Complement Color Key

Routines for performing complement color key replacement.

CompColorKey

Complement color key replacement.

NppStatus **nppiCompColorKey_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nColorKeyConst, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

Parameters

- ▶ **pSrc1** – source1 packed pixel format image pointer.
- ▶ **nSrc1Step** – source1 packed pixel format image line step.
- ▶ **pSrc2** – source2 packed pixel format image pointer.
- ▶ **nSrc2Step** – source2 packed pixel format image line step.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nColorKeyConst** – color key constant
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCompColorKey_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nColorKeyConst[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

Parameters

- ▶ **pSrc1** – source1 packed pixel format image pointer.
- ▶ **nSrc1Step** – source1 packed pixel format image line step.
- ▶ **pSrc2** – source2 packed pixel format image pointer.
- ▶ **nSrc2Step** – source2 packed pixel format image line step.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nColorKeyConst** – color key constant array
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCompColorKey_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nColorKeyConst[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

Parameters

- ▶ **pSrc1** – source1 packed pixel format image pointer.
- ▶ **nSrc1Step** – source1 packed pixel format image line step.
- ▶ **pSrc2** – source2 packed pixel format image pointer.
- ▶ **nSrc2Step** – source2 packed pixel format image line step.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nColorKeyConst** – color key constant array
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiAlphaCompColorKey_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, *Npp8u* nAlpha1, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* nAlpha2, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nColorKeyConst[4], *NppiAlphaOp* nppAlphaOp, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2 with alpha blending.

Parameters

- ▶ **pSrc1** – source1 packed pixel format image pointer.
- ▶ **nSrc1Step** – source1 packed pixel format image line step.
- ▶ **nAlpha1** – source1 image alpha opacity (0 - max channel pixel value).
- ▶ **pSrc2** – source2 packed pixel format image pointer.
- ▶ **nSrc2Step** – source2 packed pixel format image line step.
- ▶ **nAlpha2** – source2 image alpha opacity (0 - max channel pixel value).
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nColorKeyConst** – color key constant array
- ▶ **nppAlphaOp** – NppiAlphaOp alpha compositing operation selector (excluding premul ops).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.1.5 ColorTwist

Routines for converting between various image color models using user supplied matrix coefficients.

ColorTwist

Perform color twist pixel processing.

Color twist consists of applying the following formula to each image pixel using coefficients from the user supplied color twist host matrix array as follows where dst[x] and src[x] represent destination pixel and source pixel channel or plane x. The full sized coefficient matrix should be sent for all pixel channel sizes, the function will process the appropriate coefficients and channels for the corresponding pixel size.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +  
↪ aTwist[0][3]  
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +  
↪ aTwist[1][3]  
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +  
↪ aTwist[2][3]
```

NppStatus **nppiColorTwist32f_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.

- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_C2IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_C3IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned color twist, with alpha copy.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is copied unmodified from the source pixel to the destination pixel.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_C4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is unmodified.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_AC4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwist32fC_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[4][4], const *Npp32f* aConstants[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned color twist with 4x4 matrix and constant vector addition.

An input 4x4 color twist matrix with floating-point coefficient values with an additional constant vector addition is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪ aTwist[0][3] * src[3] + aConstants[0]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪ aTwist[1][3] * src[3] + aConstants[1]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪ aTwist[2][3] * src[3] + aConstants[2]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪ aTwist[3][3] * src[3] + aConstants[3]
```

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **aConstants** – fixed size array of constant values, one per channel..
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwist32fC_8u_C4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[4][4], const *Npp32f* aConstants[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned in place color twist with 4x4 matrix and an additional constant vector addition.

An input 4x4 color twist matrix with floating-point coefficient values with an additional constant vector addition is applied within ROI. For this particular version of the function the result is generated as shown below.

```

dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪aTwist[0][3] * src[3] + aConstants[0]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪aTwist[1][3] * src[3] + aConstants[1]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪aTwist[2][3] * src[3] + aConstants[2]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪aTwist[3][3] * src[3] + aConstants[3]

```

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **aConstants** – fixed size array of constant values, one per channel..
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *const pDst[3], int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8u_IP3R_Ctx**(*Npp8u* *const pSrcDst[3], int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place planar pixel format image pointer array, one pointer per plane.
- ▶ **nSrcDstStep** – in place planar pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_C1R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 8-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_C1IR_Ctx**(*Npp8s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 8-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_C2R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 8-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_C2IR_Ctx**(*Npp8s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 8-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 8-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_C3IR_Ctx**(*Npp8s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 8-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit signed color twist, with alpha copy.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is copied unmodified from the source pixel to the destination pixel.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_C4IR_Ctx**(*Npp8s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit signed in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is unmodified.

Parameters

- **pSrcDst** – in place packed pixel format image pointer.
- **nSrcDstStep** – in place packed pixel format image line step.
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_AC4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit signed color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_AC4IR_Ctx**(*Npp8s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit signed in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_P3R_Ctx**(const *Npp8s* *const pSrc[3], int nSrcStep, *Npp8s* *const pDst[3], int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 8-bit signed planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_8s_IP3R_Ctx**(*Npp8s* *const pSrcDst[3], int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 8-bit signed planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place planar pixel format image pointer array, one pointer per plane.
- ▶ **nSrcDstStep** – in place planar pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16u_C2R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 16-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16u_C2IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 16-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16u_C3IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16u_AC4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16u_P3R_Ctx**(const *Npp16u* *const pSrc[3], int nSrcStep, *Npp16u* *const pDst[3], int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16u_IP3R_Ctx**(*Npp16u* *const pSrcDst[3], int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place planar pixel format image pointer array, one pointer per plane.
- ▶ **nSrcDstStep** – in place planar pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16s_C1IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16s_C2R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 16-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16s_C2IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 16-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.

- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16s_C3IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16s_AC4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16s_P3R_Ctx**(const *Npp16s* *const pSrc[3], int nSrcStep, *Npp16s* *const pDst[3], int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16s_IP3R_Ctx**(*Npp16s* *const pSrcDst[3], int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place planar pixel format image pointer array, one pointer per plane.
- ▶ **nSrcDstStep** – in place planar pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16f_C1R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 16-bit floating point color twist.

An input color twist matrix with 32-bit floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16f_C1IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 16-bit floating point in place color twist.

An input color twist matrix with 32-bit floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16f_C2R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 16-bit floating point color twist.

An input color twist matrix with 32-bit floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16f_C2IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 16-bit floating point in place color twist.

An input color twist matrix with 32-bit floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16f_C3R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 16-bit floating point color twist.

An input color twist matrix with 32-bit floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.

- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16f_C3IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 16-bit floating point in place color twist.

An input color twist matrix with 32-bit floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32f_16f_C4R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point color twist, with alpha copy.

An input color twist matrix with 32-bit floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is copied unmodified from the source pixel to the destination pixel.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwist32f_16f_C4IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point in place color twist, not affecting Alpha.

An input color twist matrix with 32-bit floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not modified.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwist32fc_16f_C4R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[4][4], const *Npp32f* aConstants[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point color twist with 4x4 matrix and constant vector addition.

An input 4x4 color twist matrix with 32-bit floating-point coefficient values with an additional 32-bit floating point constant vector addition is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪ aTwist[0][3] * src[3] + aConstants[0]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪ aTwist[1][3] * src[3] + aConstants[1]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪ aTwist[2][3] * src[3] + aConstants[2]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪ aTwist[3][3] * src[3] + aConstants[3]
```

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- **aConstants** – fixed size array of constant values, one per channel..
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist32fC_16f_C4IR_Ctx**(*Npp16f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[4][4], const *Npp32f* aConstants[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point in place color twist with 4x4 matrix and an additional constant vector addition.

An input 4x4 color twist matrix with 32-bit floating-point coefficient values with an additional 32-bit floating point constant vector addition is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪ aTwist[0][3] * src[3] + aConstants[0]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪ aTwist[1][3] * src[3] + aConstants[1]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪ aTwist[2][3] * src[3] + aConstants[2]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪ aTwist[3][3] * src[3] + aConstants[3]
```

Parameters

- **pSrcDst** – in place packed pixel format image pointer.
- **nSrcDstStep** – in place packed pixel format image line step.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- **aConstants** – fixed size array of constant values, one per channel..
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_C1IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_C2R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 32-bit floating point color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_C2IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

2 channel 32-bit floating point in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point color twist, with alpha copy.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is copied unmodified from the source pixel to the destination pixel.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_C4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not modified.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32fC_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[4][4], const *Npp32f* aConstants[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point color twist with 4x4 matrix and constant vector addition.

An input 4x4 color twist matrix with floating-point coefficient values with an additional constant vector addition is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪aTwist[0][3] * src[3] + aConstants[0]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪aTwist[1][3] * src[3] + aConstants[1]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪aTwist[2][3] * src[3] + aConstants[2]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪aTwist[3][3] * src[3] + aConstants[3]
```

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **aConstants** – fixed size array of constant values, one per channel..
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32fC_4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[4][4], const *Npp32f* aConstants[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place color twist with 4x4 matrix and an additional constant vector addition.

An input 4x4 color twist matrix with floating-point coefficient values with an additional constant vector addition is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪aTwist[0][3] * src[3] + aConstants[0]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪aTwist[1][3] * src[3] + aConstants[1]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪aTwist[2][3] * src[3] + aConstants[2]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪aTwist[3][3] * src[3] + aConstants[3]
```

Parameters

- ▶ **pSrcDst** – in place packed pixel format image pointer.
- ▶ **nSrcDstStep** – in place packed pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **aConstants** – fixed size array of constant values, one per channel..
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_P3R_Ctx**(const *Npp32f* *const pSrc[3], int nSrcStep, *Npp32f* *const pDst[3], int nDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwist_32f_IP3R_Ctx**(*Npp32f* *const pSrcDst[3], int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* aTwist[3][4], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

Parameters

- ▶ **pSrcDst** – in place planar pixel format image pointer array, one pointer per plane.
- ▶ **nSrcDstStep** – in place planar pixel format image line step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aTwist** – Host memory array containing the color twist matrix with floating-point coefficient values.

► **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.1.6 ColorTwistBatch

Routines for converting between various image color models using user supplied matrix coefficients on batches of images.

ColorTwistBatch

Perform color twist pixel batch processing.

Color twist consists of applying the following formula to each image pixel using coefficients from one or more user supplied color twist device memory matrix arrays as follows where dst[x] and src[x] represent destination pixel and source pixel channel or plane x. The full sized coefficient matrix should be sent for all pixel channel sizes, the function will process the appropriate coefficients and channels for the corresponding pixel size. ColorTwistBatch generally takes the same parameter list as ColorTwist except that there is a list of N instances of those parameters (N > 1) and that list is passed in device memory; The matrix pointers referenced for each image in the batch also need to point to device memory matrix values. A convenient data structure is provided that allows for easy initialization of the parameter lists. The only restriction on these functions is that there is one single ROI which is applied respectively to each image in the batch. The primary purpose of this function is to provide improved performance for batches of smaller images as long as GPU resources are available. Therefore it is recommended that the function not be used for very large images as there may not be resources available for processing several large images simultaneously.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪ aTwist[0][3]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪ aTwist[1][3]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪ aTwist[2][3]
```

NppStatus **nppiColorTwistBatch32f_8u_C1R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer color twist batch.

An input color twist matrix with floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- **nMin** – Minimum clamp value.
- **nMax** – Maximum saturation and clamp value.
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwist-BatchCXR* structures.

- **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32f_8u_C1IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer in place color twist batch.

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- **nMin** – Minimum clamp value.
- **nMax** – Maximum saturation and clamp value.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32f_8u_C3R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer color twist batch.

An input color twist matrix with floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- **nMin** – Minimum clamp value.
- **nMax** – Maximum saturation and clamp value.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwistBatch32f_8u_C3IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer in place color twist batch.

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwistBatch32f_8u_C4R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer color twist batch.

An input color twist matrix with floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

```
NppStatus nppiColorTwistBatch32f_8u_C4IR_Ctx(Npp32f nMin, Npp32f nMax, NppiSize
                                             oSizeROI, NppiColorTwistBatchCXR
                                             *pBatchList, int nBatchSize,
                                             NppStreamContext nppStreamCtx)
```

4 channel 8-bit unsigned integer in place color twist batch.

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiColorTwistBatch32f_8u_AC4R_Ctx(Npp32f nMin, Npp32f nMax, NppiSize
                                             oSizeROI, NppiColorTwistBatchCXR
                                             *pBatchList, int nBatchSize,
                                             NppStreamContext nppStreamCtx)
```

4 channel 8-bit unsigned integer color twist batch, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiColorTwistBatch32f_8u_AC4IR_Ctx(Npp32f nMin, Npp32f nMax, NppiSize
                                             oSizeROI, NppiColorTwistBatchCXR
                                             *pBatchList, int nBatchSize,
                                             NppStreamContext nppStreamCtx)
```

4 channel 8-bit unsigned integer in place color twist batch, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32fC_8u_C4R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer color twist with 4x5 matrix including a constant vector (20 coefficients total).

An input 4x5 color twist matrix with floating-point coefficient values including a constant (in the fourth column) vector is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪ aTwist[0][3] * src[3] + aTwist[0][4]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪ aTwist[1][3] * src[3] + aTwist[1][4]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪ aTwist[2][3] * src[3] + aTwist[2][4]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪ aTwist[3][3] * src[3] + aTwist[3][4]
```

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.

- **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwistBatch32fC_8u_C4IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer in place color twist with 4x5 matrix including a constant vector (20 coefficients total).

An input 4x5 color twist matrix with floating-point coefficient values including a constant (in the fourth column) vector is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪ aTwist[0][3] * src[3] + aTwist[0][4]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪ aTwist[1][3] * src[3] + aTwist[1][4]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪ aTwist[2][3] * src[3] + aTwist[2][4]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪ aTwist[3][3] * src[3] + aTwist[3][4]
```

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- **nMin** – Minimum clamp value.
- **nMax** – Maximum saturation and clamp value.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwistBatch_32f_C1R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point color twist batch.

An input color twist matrix with floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch_32f_C1IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point in place color twist batch.

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch_32f_C3R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point color twist batch.

An input color twist matrix with floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch_32f_C3IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point in place color twist batch.

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch_32f_C4R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point color twist batch.

An input color twist matrix with floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwistBatch_32f_C4IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place color twist batch.

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwistBatch_32f_AC4R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point color twist batch, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiColorTwistBatch_32f_AC4IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place color twist batch, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- **nMin** – Minimum clamp value.
- **nMax** – Maximum saturation and clamp value.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch_32fc_C4R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point color twist with 4x5 matrix including a constant vector (20 coefficients total).

An input 4x5 color twist matrix with floating-point coefficient values including a constant (in the fourth column) vector is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪ aTwist[0][3] * src[3] + aTwist[0][4]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪ aTwist[1][3] * src[3] + aTwist[1][4]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪ aTwist[2][3] * src[3] + aTwist[2][4]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪ aTwist[3][3] * src[3] + aTwist[3][4]
```

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- **nMin** – Minimum clamp value.
- **nMax** – Maximum saturation and clamp value.
- **oSizeROI** – *Region-Of-Interest (ROI)*.

- **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch_32fC4IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel in place 32-bit floating point color twist with 4x5 matrix including a constant vector (20 coefficients total).

An input 4x5 color twist matrix with floating-point coefficient values including a constant (in the fourth column) vector is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪ aTwist[0][3] * src[3] + aTwist[0][4]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪ aTwist[1][3] * src[3] + aTwist[1][4]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪ aTwist[2][3] * src[3] + aTwist[2][4]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪ aTwist[3][3] * src[3] + aTwist[3][4]
```

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- **nMin** – Minimum clamp value.
- **nMax** – Maximum saturation and clamp value.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32f_16f_C1R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 16-bit floating point color twist batch.

An input color twist matrix with 32-bit floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32f_16f_C1IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 16-bit floating point in place color twist batch.

An input color twist matrix with 32-bit floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32f_16f_C3R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 16-bit floating point color twist batch.

An input color twist matrix with 32-bit floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32f_16f_C3IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 16-bit floating point in place color twist batch.

An input color twist matrix with floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32f_16f_C4R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point color twist batch.

An input color twist matrix with 32-bit floating-point coefficient values is applied within the ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.

- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32f_16f_C4IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point in place color twist batch.

An input color twist matrix with 32-bit floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- ▶ **nMin** – Minimum clamp value.
- ▶ **nMax** – Maximum saturation and clamp value.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32fC_16f_C4R_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point color twist with 4x5 matrix including a constant vector (20 coefficients total).

An input 4x5 color twist matrix with 32-bit floating-point coefficient values including a constant (in the fourth column) vector is applied within ROI. For this particular version of the function the result is generated as shown below.

```
dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪ aTwist[0][3] * src[3] + aTwist[0][4]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪ aTwist[1][3] * src[3] + aTwist[1][4]
```

(continues on next page)

(continued from previous page)

```

dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪aTwist[2][3] * src[3] + aTwist[2][4]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪aTwist[3][3] * src[3] + aTwist[3][4]

```

An input color twist matrix with 32-bit floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- **nMin** – Minimum clamp value.
- **nMax** – Maximum saturation and clamp value.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiColorTwistBatch32fc_16f_C4IR_Ctx**(*Npp32f* nMin, *Npp32f* nMax, *NppiSize* oSizeROI, *NppiColorTwistBatchCXR* *pBatchList, int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel in place 16-bit floating point color twist with 4x5 matrix including a constant vector (20 coefficients total).

An input 4x5 color twist matrix with 32-bit floating-point coefficient values including a constant (in the fourth column) vector is applied within ROI. For this particular version of the function the result is generated as shown below.

```

dst[0] = aTwist[0][0] * src[0] + aTwist[0][1] * src[1] + aTwist[0][2] * src[2] +
↪aTwist[0][3] * src[3] + aTwist[0][4]
dst[1] = aTwist[1][0] * src[0] + aTwist[1][1] * src[1] + aTwist[1][2] * src[2] +
↪aTwist[1][3] * src[3] + aTwist[1][4]
dst[2] = aTwist[2][0] * src[0] + aTwist[2][1] * src[1] + aTwist[2][2] * src[2] +
↪aTwist[2][3] * src[3] + aTwist[2][4]
dst[3] = aTwist[3][0] * src[0] + aTwist[3][1] * src[1] + aTwist[3][2] * src[2] +
↪aTwist[3][3] * src[3] + aTwist[3][4]

```

An input color twist matrix with 32-bit floating-point coefficient values is applied within ROI for each image in batch. Color twist matrix can vary per image. The same ROI is applied to each image.

Parameters

- **nMin** – Minimum clamp value.
- **nMax** – Maximum saturation and clamp value.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiColorTwistBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiColorTwistBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.1.7 ColorLUT

Perform image color processing using members of various types of color look up tables.

Functions

NppStatus **nppiLUT_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

8-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_8u_C3IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_8u_C4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_8u_AC4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16u_C3IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16u_C4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16u_AC4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16s_C1IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16s_C3IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16s_C4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_16s_AC4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues, const *Npp32f* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_32f_C1IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues, const *Npp32f* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- **pSrcDst** – *In-Place Image Pointer*.
- **nSrcDstStep** – *In-Place-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[4], const *Npp32f* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_32f_C4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[4], const *Npp32f* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

1.8.1.8 ColorLUTLinear

Perform image color processing using linear interpolation between members of various types of color look up tables.

Functions

NppStatus **nppiLUT_Linear_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

8-bit unsigned linear interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

```
* >>>>>> ATTENTION ATTENTION <<<<<<<
*
```

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be device memory pointers.

```
* >>>>>> <<<<<<<
*
```

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is now a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is now a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Linear_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

8-bit unsigned linear interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Linear_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned linear interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

```
* >>>>>> ATTENTION ATTENTION <<<<<<
*
```

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be host memory pointers to arrays of device memory pointers.

```
* >>>>>> <<<<<<
*
```

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Linear_8u_C3IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned linear interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Linear_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned linear interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

```
* >>>>>> ATTENTION ATTENTION <<<<<<<
*
```

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be host memory pointers to arrays of device memory pointers.

```
* >>>>>>>          <<<<<<<<
*
```

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Linear_8u_C4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned linear interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Linear_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned linear interpolated look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through linear interpolation. Alpha channel is the last channel and is not processed.

```
* >>>>>> ATTENTION ATTENTION <<<<<<
*
```

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be host memory pointers to arrays of device memory pointers.

```
* >>>>>> <<<<<<
*
```

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Linear_8u_AC4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned linear interpolated look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through linear interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Linear_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16u_C3IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16u_C4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16u_AC4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16s_C1IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16s_C3IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

```
NppStatus nppiLUT_Linear_16s_C4R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int
                                     nDstStep, NppiSize oSizeROI, const Npp32s
                                     *pValues[4], const Npp32s *pLevels[4], int nLevels[4],
                                     NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

```
NppStatus nppiLUT_Linear_16s_C4IR_Ctx(Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI,
                                       const Npp32s *pValues[4], const Npp32s *pLevels[4],
                                       int nLevels[4], NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_16s_AC4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues, const *Npp32f* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_32f_C1IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues, const *Npp32f* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[4], const *Npp32f* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_32f_C4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[4], const *Npp32f* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Linear_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

1.8.1.9 ColorLUTCubic

Perform image color processing using linear interpolation between members of various types of color look up tables.

Functions

NppStatus **nppiLUT_Cubic_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

8-bit unsigned cubic interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Cubic_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

8-bit unsigned cubic interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Cubic_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned cubic interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Cubic_8u_C3IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Cubic_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned cubic interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Cubic_8u_C4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Cubic_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned cubic interpolated look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through cubic interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Cubic_8u_AC4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through cubic interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Cubic_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16u_C3IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

- **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16u_C4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[4], const *Npp32s* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16u_AC4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16s_C1IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues, const *Npp32s* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16s_C3IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

```
NppStatus nppiLUT_Cubic_16s_C4R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32s *pValues[4],  
const Npp32s *pLevels[4], int nLevels[4],  
NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

```
NppStatus nppiLUT_Cubic_16s_C4IR_Ctx(Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI,  
const Npp32s *pValues[4], const Npp32s *pLevels[4],  
int nLevels[4], NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_16s_AC4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32s* *pValues[3], const *Npp32s* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues, const *Npp32f* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_32f_C1IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues, const *Npp32f* *pLevels, int nLevels, *NppStreamContext* nppStreamCtx)

32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Pointer to an array of user defined OUTPUT values (this is a device memory pointer)
- ▶ **pLevels** – Pointer to an array of user defined INPUT values (this is a device memory pointer)
- ▶ **nLevels** – Number of user defined number of input/output mapping points (levels)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[4], const *Npp32f* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_32f_C4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[4], const *Npp32f* *pLevels[4], int nLevels[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

NppStatus **nppiLUT_Cubic_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* *pValues[3], const *Npp32f* *pLevels[3], int nLevels[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- ▶ **pLevels** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- ▶ **nLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 1024 (the current size limit).

1.8.1.10 ColorLUTTrilinear

Perform image color processing using 3D trilinear interpolation between members of various types of color look up tables.

Functions

NppStatus **nppiLUT_Trilinear_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32u* *pValues, *Npp8u* *pLevels[3], int aLevels[3], *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned 3D trilinear interpolated look-up-table color conversion, with alpha copy.

Alpha channel is the last channel and is copied to the destination unmodified.

The LUT is derived from a set of user defined mapping points through trilinear interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Device pointer to aLevels[2] number of contiguous 2D x,y planes of 4-byte packed RGBX values containing the user defined base OUTPUT values at that x,y, and z (R,G,B) level location. Each level must contain x * y 4-byte packed pixel values (4th byte is used for alignment only and is ignored) in row (x) order.
- ▶ **pLevels** – Host pointer to an array of 3 host pointers, one per cube edge, pointing to user defined INPUT level values.
- ▶ **aLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per 3D cube edge. aLevels[0] represents the number of x axis levels (Red), aLevels[1] represents the number of y axis levels (Green), and aLevels[2] represents the number of z axis levels (Blue).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ **NPP_LUT_NUMBER_OF_LEVELS_ERROR** if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Trilinear_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32u* *pValues, *Npp8u* *pLevels[3], int aLevels[3], *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned 3D trilinear interpolated look-up-table color conversion, not affecting alpha.

Alpha channel is the last channel and is not processed.

The LUT is derived from a set of user defined mapping points through trilinear interpolation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Device pointer to aLevels[2] number of contiguous 2D x,y planes of 4-byte packed RGBX values containing the user defined base OUTPUT values at that x,y, and z (R,G,B) level location. Each level must contain x * y 4-byte packed pixel values (4th byte is used for alignment only and is ignored) in row (x) order.
- ▶ **pLevels** – Host pointer to an array of 3 host pointers, one per cube edge, pointing to user defined INPUT level values.

- ▶ **aLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per 3D cube edge. aLevels[0] represents the number of x axis levels (Red), aLevels[1] represents the number of y axis levels (Green), and aLevels[2] represents the number of z axis levels (Blue).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

NppStatus **nppiLUT_Trilinear_8u_AC4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *Npp32u* *pValues, *Npp8u* *pLevels[3], int aLevels[3], *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned 3D trilinear interpolated look-up-table in place color conversion, not affecting alpha.

Alpha channel is the last channel and is not processed.

The LUT is derived from a set of user defined mapping points through trilinear interpolation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pValues** – Device pointer aLevels[2] number of contiguous 2D x,y planes of 4-byte packed RGBX values containing the user defined base OUTPUT values at that x,y, and z (R,G,B) level location. Each level must contain x * y 4-byte packed pixel values (4th byte is used for alignment only and is ignored) in row (x) order.
- ▶ **pLevels** – Host pointer to an array of 3 host pointers, one per cube edge, pointing to user defined INPUT level values.
- ▶ **aLevels** – Host pointer to an array of 3 user defined number of input/output mapping points, one per 3D cube edge. aLevels[0] represents the number of x axis levels (Red), aLevels[1] represents the number of y axis levels (Green), and aLevels[2] represents the number of z axis levels (Blue).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

- ▶ NPP_LUT_NUMBER_OF_LEVELS_ERROR if the number of levels is less than 2 or greater than 256.

1.8.1.11 ColorLUTPalette

Perform image color processing using various types of bit range restricted palette color look up tables.

Functions

NppStatus **nppiLUTPalette_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pTable, int nBitSize, *NppStreamContext* nppStreamCtx)

One channel 8-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pTable** – Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 8.

NppStatus **nppiLUTPalette_8u24u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pTable, int nBitSize, *NppStreamContext* nppStreamCtx)

One channel 8-bit unsigned bit range restricted 24-bit palette look-up-table color conversion with 24-bit destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step* (3 bytes per pixel).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- ▶ **pTable** – Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 8.

NppStatus **nppiLUTPalette_8u32u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32u* *pTable, int nBitSize, *NppStreamContext* nppStreamCtx)

One channel 8-bit unsigned bit range restricted 32-bit palette look-up-table color conversion with 32-bit destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step* (4 bytes per pixel).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pTable** – Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 8.

NppStatus **nppiLUTPalette_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pTables[3], int nBitSize, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pTables** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 8.

NppStatus **nppiLUTPalette_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pTables[4], int nBitSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pTables** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 8.

NppStatus **nppiLUTPalette_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pTables[3], int nBitSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned bit range restricted palette look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pTables** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 8.

NppStatus **nppiLUTPalette_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* *pTable, int nBitSize, *NppStreamContext* nppStreamCtx)

One channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pTable** – Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 16.

NppStatus **nppiLUTPalette_16u8u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pTable, int nBitSize, *NppStreamContext* nppStreamCtx)

One channel 16-bit unsigned bit range restricted 8-bit unsigned palette look-up-table color conversion with 8-bit unsigned destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step* (1 unsigned byte per pixel).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pTable** – Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 16.

NppStatus **nppiLUTPalette_16u24u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pTable, int nBitSize, *NppStreamContext* nppStreamCtx)

One channel 16-bit unsigned bit range restricted 24-bit unsigned palette look-up-table color conversion with 24-bit unsigned destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step* (3 unsigned bytes per pixel).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pTable** – Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 16.

NppStatus **nppiLUTPalette_16u32u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32u* *pTable, int nBitSize, *NppStreamContext* nppStreamCtx)

One channel 16-bit unsigned bit range restricted 32-bit palette look-up-table color conversion with 32-bit unsigned destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step* (4 bytes per pixel).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pTable** – Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 16.

NppStatus **nppiLUTPalette_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* *pTables[3], int nBitSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pTables** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 16.

NppStatus **nppiLUTPalette_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* *pTables[4], int nBitSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pTables** – Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.
- **nBitSize** – Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 16.

NppStatus **nppiLUTPalette_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* *pTables[3], int nBitSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned bit range restricted palette look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. Alpha channel is the last channel and is not processed.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pTables** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.

- **nBitSize** – Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 16.

NppStatus **nppiLUTPaletteSwap_8u_C3A0C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, int nAlphaValue, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pTables[3], int nBitSize, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned source bit range restricted palette look-up-table color conversion to four channel 8-bit unsigned destination output with alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. This function also reverses the source pixel channel order in the destination so the Alpha channel is the first channel.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step* (3 bytes per pixel).
- **nAlphaValue** – Signed alpha value that will be used to initialize the pixel alpha channel position in all modified destination pixels.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step* (4 bytes per pixel with alpha).
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pTables** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values. Alpha values < 0 or > 255 will cause destination pixel alpha channel values to be unmodified.
- **nBitSize** – Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 8.

NppStatus **nppiLUTPaletteSwap_16u_C3A0C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, int nAlphaValue, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* *pTables[3], int nBitSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned source bit range restricted palette look-up-table color conversion to four channel 16-bit unsigned destination output with alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. This function also

reverses the source pixel channel order in the destination so the Alpha channel is the first channel.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step* (3 unsigned short integers per pixel).
- ▶ **nAlphaValue** – Signed alpha value that will be used to initialize the pixel alpha channel position in all modified destination pixels.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step* (4 unsigned short integers per pixel with alpha).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pTables** – Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values. Alpha values < 0 or > 65535 will cause destination pixel alpha channel values to be unmodified.
- ▶ **nBitSize** – Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

- ▶ NPP_LUT_PALETTE_BITSIZE_ERROR if nBitSize is < 1 or > 16.

1.8.2. Color Sampling Format Conversion Functions

Routines for converting between various image color sampling formats.

1.8.2.1 YCbCr420ToYCbCr411

YCbCr420 to YCbCr411 sampling format conversion.

Functions

NppStatus **nppiYCbCr420ToYCbCr411_8u_P3P2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDstY, int nDstYStep, *Npp8u* *pDstCbCr, int nDstCbCrStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array*.

- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*
- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*
- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr420ToYCbCr411_8u_P2P3R_Ctx**(const *Npp8u* *pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer.*
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step.*
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer.*
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.2 YCbCr422ToYCbCr422

YCbCr422 to YCbCr422 sampling format conversion.

Functions

NppStatus **nppiYCbCr422_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422_8u_P3C2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array*.
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.3 YCbCr422ToYCrCb422

YCbCr422 to YCrCb422 sampling format conversion.

Functions

NppStatus **nppiYCbCr422ToYCrCb422_8u_C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToYCrCb422_8u_P3C2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.4 YCbCr422ToCbYCr422

YCbCr422 to CbYCr422 sampling format conversion.

Functions

NppStatus **nppiYCbCr422ToCbYCr422_8u_C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.5 CbYCr422ToYCbCr411

CbYCr422 to YCbCr411 sampling format conversion.

Functions

NppStatus **nppiCbYCr422ToYCbCr411_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.6 YCbCr422ToYCbCr420

YCbCr422 to YCbCr420 sampling format conversion.

Functions

NppStatus **nppiYCbCr422ToYCbCr420_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst[3], int nDstStep[3], *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToYCbCr420_8u_P3P2R_Ctx**(const *Npp8u* *const pSrc[3], int
rSrcStep[3], *Npp8u* *pDstY, int nDstYStep,
Npp8u *pDstCbCr, int nDstCbCrStep,
NppiSize oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*
- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*

- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToYCbCr420_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToYCbCr420_8u_C2P2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDstY, int nDstYStep, *Npp8u* *pDstCbCr, int nDstCbCrStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*
- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*
- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.7 YCrCb420ToYCbCr422

YCrCb420 to YCbCr422 sampling format conversion.

Functions

NppStatus **nppiYCrCb420ToYCbCr422_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst[3], int rDstStep[3], *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCrCb420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCrCb420ToYCbCr422_8u_P3C2R_Ctx**(const *Npp8u* *const pSrc[3], int
rSrcStep[3], *Npp8u* *pDst, int nDstStep,
NppiSize oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.8 YCbCr422ToYCrCb420

YCbCr422 to YCrCb420 sampling format conversion.

Functions

NppStatus **nppiYCbCr422ToYCrCb420_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.9 YCbCr422ToYCbCr411

YCbCr422 to YCbCr411 sampling format conversion.

Functions

NppStatus **nppiYCbCr422ToYCbCr411_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array*.
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToYCbCr411_8u_P3P2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDstY, int nDstYStep, *Npp8u* *pDstCbCr, int nDstCbCrStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*
- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*
- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToYCbCr411_8u_G2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToYCbCr411_8u_C2P2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDstY, int nDstYStep, *Npp8u* *pDstCbCr, int nDstCbCrStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*
- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*
- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.10 YCrCb422ToYCbCr422

YCrCb422 to YCbCr422 sampling format conversion.

Functions

NppStatus **nppiYCrCb422ToYCbCr422_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.11 YCrCb422ToYCbCr420

YCrCb422 to YCbCr420 sampling format conversion.

Functions

NppStatus **nppiYCrCb422ToYCbCr420_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.12 YCrCb422ToYCbCr411

YCrCb422 to YCbCr411 sampling format conversion.

Functions

NppStatus **nppiYCrCb422ToYCbCr411_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.13 CbYCr422ToYCbCr422

CbYCr422 to YCbCr422 sampling format conversion.

Functions

NppStatus **nppiCbYCr422ToYCbCr422_8u_C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCbYCr422ToYCbCr422_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.14 CbYCr422ToYCbCr420

CbYCr422 to YCbCr420 sampling format conversion.

Functions

NppStatus **nppiCbYCr422ToYCbCr420_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCbYCr422ToYCbCr420_8u_C2P2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDstY, int nDstYStep, *Npp8u* *pDstCbCr, int nDstCbCrStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*

- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*
- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.15 CbYCr422ToYCrCb420

CbYCr422 to YCrCb420 sampling format conversion.

Functions

NppStatus **nppiCbYCr422ToYCrCb420_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.16 YCbCr420ToYCbCr420

YCbCr420 to YCbCr420 sampling format conversion.

Functions

NppStatus **nppiYCbCr420_8u_P3P2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDstY, int nDstYStep, *Npp8u* *pDstCbCr, int nDstCbCrStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*
- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*
- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr420_8u_P2P3R_Ctx**(const *Npp8u* *const pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer.*
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step.*
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer.*
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.17 YCbCr420ToYCbCr422

YCbCr420 to YCbCr422 sampling format conversion.

Functions

NppStatus **nppiYCbCr420ToYCbCr422_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst[3], int nDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr420ToYCbCr422_8u_P2P3R_Ctx**(const *Npp8u* *pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer.*
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step.*
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer.*
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr420ToYCbCr422_8u_P2C2R_Ctx**(const *Npp8u* *pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer.*
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step.*
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer.*
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.18 YCbCr420ToCbYCr422

YCbCr420 to CbYCr422 sampling format conversion.

Functions

NppStatus **nppiYCbCr420ToCbYCr422_8u_P2C2R_Ctx**(const *Npp8u* *pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer.*
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step.*
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer.*
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes***1.8.2.19 YCbCr420ToYCrCb420**

YCbCr420 to YCrCb420 sampling format conversion.

Functions

NppStatus **nppiYCbCr420ToYCrCb420_8u_P2P3R_Ctx**(const *Npp8u* *pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer.*
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step.*
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer.*
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes***1.8.2.20 YCrCb420ToCbYCr422**

YCrCb420 to CbYCr422 sampling format conversion.

Functions

NppStatus **nppiYCrCb420ToCbYCr422_8u_P3C2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.21 YCrCb420ToYCbCr420

YCrCb420 to YCbCr420 sampling format conversion.

Functions

NppStatus **nppiYCrCb420ToYCbCr420_8u_P3P2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDstY, int nDstYStep, *Npp8u* *pDstCbCr, int nDstCbCrStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*
- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*
- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.22 YCrCb420ToYCbCr411

YCrCb420 to YCbCr411 sampling format conversion.

Functions

NppStatus **nppiYCrCb420ToYCbCr411_8u_P3P2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDstY, int nDstYStep, *Npp8u* *pDstCbCr, int nDstCbCrStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*
- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*
- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.23 YCbCr411ToYCbCr411

YCbCr411 to YCbCr411 sampling format conversion.

Functions

NppStatus **nppiYCbCr411_8u_P3P2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDstY, int nDstYStep, *Npp8u* *pDstCbCr, int nDstCbCrStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*

- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*
- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*
- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411_8u_P2P3R_Ctx**(const *Npp8u* *pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer.*
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step.*
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer.*
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.24 YCbCr411ToYCbCr422

YCbCr411 to YCbCr422 sampling format conversion.

Functions

NppStatus **nppiYCbCr411ToYCbCr422_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst[3], int nDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToYCbCr422_8u_P3C2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToYCbCr422_8u_P2P3R_Ctx**(const *Npp8u* *const pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer.*
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step.*
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer.*
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*

- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToYCbCr422_8u_P2C2R_Ctx**(const *Npp8u* *pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer*.
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step*.
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer*.
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.25 YCbCr411ToYCrCb422

YCbCr411 to YCrCb422 sampling format conversion.

Functions

NppStatus **nppiYCbCr411ToYCrCb422_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst[3], int nDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb422 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array*.
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.

- **nDstStep** – Destination-Planar-Image Line Step Array.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToYCrCb422_8u_P3C2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array*.
- **rSrcStep** – *Source-Planar-Image Line Step Array*.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.2.26 YCbCr411ToYCbCr420

YCbCr411 to YCbCr420 sampling format conversion.

Functions

NppStatus **nppiYCbCr411ToYCbCr420_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst[3], int nDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array*.
- **rSrcStep** – *Source-Planar-Image Line Step Array*.
- **pDst** – *Destination-Planar-Image Pointer Array*.
- **nDstStep** – Destination-Planar-Image Line Step Array.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToYCbCr420_8u_P3P2R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDstY, int nDstYStep, *Npp8u* *pDstCbCr, int nDstCbCrStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDstY** – *Destination-Planar-Image Pointer.*
- ▶ **nDstYStep** – *Destination-Planar-Image Line Step.*
- ▶ **pDstCbCr** – *Destination-Planar-Image Pointer.*
- ▶ **nDstCbCrStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToYCbCr420_8u_P2P3R_Ctx**(const *Npp8u* *pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer.*
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step.*
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer.*
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes***1.8.2.27 YCbCr411ToYCrCb420**

YCbCr411 to YCrCb420 sampling format conversion.

Functions

NppStatus **nppiYCbCr411ToYCrCb420_8u_P2P3R_Ctx**(const *Npp8u* *pSrcY, int nSrcYStep, const *Npp8u* *pSrcCbCr, int nSrcCbCrStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

Parameters

- ▶ **pSrcY** – *Source-Planar-Image Pointer*.
- ▶ **nSrcYStep** – *Source-Planar-Image Line Step*.
- ▶ **pSrcCbCr** – *Source-Planar-Image Pointer*.
- ▶ **nSrcCbCrStep** – *Source-Planar-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes***1.8.2.28 NV12ToYUV420**

NV12 to YUV420 color conversion.

Functions

NppStatus **nppiNV12ToYUV420_8u_P2P3R_Ctx**(const *Npp8u* *const pSrc[2], int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV12 to 3 channel 8-bit unsigned planar YUV420 color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for UV plane).

- ▶ **nSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **aDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3. Color Model Conversion Functions

Routines for converting between various image color models.

1.8.3.1 RGBToYUV

RGB to YUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YUV. For digital RGB values in the range [0..255], Y has the range [0..255], U varies in the range [-112..+112], and V in the range [-157..+157]. To fit in the range of [0..255], a constant value of 128 is added to computed U and V values, and V is then saturated.

```
Npp32f nY = 0.299F * R + 0.587F * G + 0.114F * B;
Npp32f nU = (0.492F * ((Npp32f)nB - nY)) + 128.0F;
Npp32f nV = (0.877F * ((Npp32f)nR - nY)) + 128.0F;
if (nV > 255.0F)
    nV = 255.0F;
```

Functions

NppStatus **nppiRGBToYUV_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YUV color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiRGBToYUV_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiRGBToYUV_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiRGBToYUV_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*

- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYUV_8u_AC4P4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[4], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha.

images.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Planar-Image Pointer Array.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.2 BGRTToYUV

BGR to YUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YUV. For digital RGB values in the range [0..255], Y has the range [0..255], U varies in the range [-112..+112], and V in the range [-157..+157]. To fit in the range of [0..255], a constant value of 128 is added to computed U and V values, and V is then saturated.

```
Npp32f nY = 0.299F * R + 0.587F * G + 0.114F * B;  
Npp32f nU = (0.492F * ((Npp32f)nB - nY)) + 128.0F;  
Npp32f nV = (0.877F * ((Npp32f)nR - nY)) + 128.0F;  
if (nV > 255.0F)  
    nV = 255.0F;
```


Functions

NppStatus **nppiBGRToYUV_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed YUV color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYUV_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYUV_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar YUV color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYUV_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YUV color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYUV_8u_AC4P4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[4], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.3 YUVToRGB

YUV to RGB color conversion.

Here is how NPP converts YUV to gamma corrected RGB or BGR.

```
Npp32f nY = (Npp32f)Y;
Npp32f nU = (Npp32f)U - 128.0F;
Npp32f nV = (Npp32f)V - 128.0F;
Npp32f nR = nY + 1.140F * nV;
if (nR < 0.0F)
    nR = 0.0F;
if (nR > 255.0F)
    nR = 255.0F;
Npp32f nG = nY - 0.394F * nU - 0.581F * nV;
if (nG < 0.0F)
    nG = 0.0F;
if (nG > 255.0F)
    nG = 255.0F;
Npp32f nB = nY + 2.032F * nU;
if (nB < 0.0F)
    nB = 0.0F;
if (nB > 255.0F)
    nB = 255.0F;
```

Functions

NppStatus **nppiYUVToRGB_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUVToRGB_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed RGB color conversion with alpha, not affecting alpha.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUVToRGB_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3],
int nDstStep, *NppiSize* oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUVToRGB_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst,
int nDstStep, *NppiSize* oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.4 YUVToRGBBatch

YUV to RGB batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

NPP converts YUV to gamma corrected RGB the same way as in *YUVToRGB*.

Functions

NppStatus **nppiYUVToRGBBatch_8u_C3R_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed RGB batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input and output images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer*.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUVToRGBBatch_8u_P3C3R_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed RGB batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of U planes. The third element of array (pSrcBatchList[2]) represents a batch of V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.

- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.5 YUVToRGBBatchAdvanced

YUV to RGB batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

NPP converts YUV to gamma corrected RGB the same way as in *YUVToRGB*.

Functions

NppStatus **nppiYUVToRGBBatch_8u_C3R_Advanced_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed RGB batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer*.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUVToRGBBatch_8u_P3C3R_Advanced_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed RGB batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of U planes. The third element of array (pSrcBatchList[2]) represents a batch of V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.6 YUVToBGR

YUV to BGR color conversion.

Here is how NPP converts YUV to gamma corrected RGB or BGR.

```
Npp32f nY = (Npp32f)Y;
Npp32f nU = (Npp32f)U - 128.0F;
Npp32f nV = (Npp32f)V - 128.0F;
Npp32f nR = nY + 1.140F * nV;
if (nR < 0.0F)
    nR = 0.0F;
if (nR > 255.0F)
    nR = 255.0F;
Npp32f nG = nY - 0.394F * nU - 0.581F * nV;
if (nG < 0.0F)
    nG = 0.0F;
if (nG > 255.0F)
    nG = 255.0F;
Npp32f nB = nY + 2.032F * nU;
if (nB < 0.0F)
    nB = 0.0F;
if (nB > 255.0F)
    nB = 255.0F;
```

Functions

NppStatus **nppiYUVToBGR_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUVToBGR_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed BGR color conversion with alpha, not affecting alpha.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUVToBGR_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*

- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUVToBGR_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.7 YUVToBGRBatch

YUV to BGR batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

NPP converts YUV to gamma corrected BGR the same way as in *YUVToBGR*.

Functions

NppStatus **nppiYUVToBGRBatch_8u_C3R_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed BGR batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input and output images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer.*
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer.*

- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiYUVToBGRBatch_8u_P3C3R_Ctx(const NppImageDescriptor *const  
pSrcBatchList[3], NppImageDescriptor  
*pDstBatchList, int nBatchSize, NppiSize  
oSizeROI, NppStreamContext nppStreamCtx)
```

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed BGR batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of U planes. The third element of array (pSrcBatchList[2]) represents a batch of V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.8 YUVToBGRBatchAdvanced

YUV to BGR batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

NPP converts YUV to gamma corrected BGR the same way as in *YUVToBGR*.

Functions

NppStatus **nppiYUVToBGRBatch_8u_C3R_Advanced_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed BGR batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer*.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUVToBGRBatch_8u_P3C3R_Advanced_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed BGR batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of U planes. The third element of array (pSrcBatchList[2]) represents a batch of V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.9 RGBToYUV422

RGB to YUV422 color conversion.

NPP converts YUV to gamma corrected BGR the same way as in *YUVToBGR*.

Functions

NppStatus **nppiRGBToYUV422_8u_C3C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YUV422 color conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYUV422_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV422 color conversion. images.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Planar-Image Pointer Array.*
- **rDstStep** – *Destination-Planar-Image Line Step Array.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYUV422_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.10 YUV422ToRGB

YUV422 to RGB color conversion.

Functions

NppStatus **nppiYUV422ToRGB_8u_C2C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV422ToRGB_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV422ToRGB_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV422ToRGB_8u_P3AC4R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV422 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.11 YUV422ToRGBBatch

Planar YUV422 to packed RGB batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

Functions

NppStatus **nppiYUV422ToRGBBatch_8u_P3C3R_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed RGB batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of U planes. The third element of array (pSrcBatchList[2]) represents a batch of V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.12 YUV422ToRGBBatchAdvanced

Planar YUV422 to packed RGB batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

Functions

NppStatus **nppiYUV422ToRGBBatch_8u_P3C3R_Advanced_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed RGB batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of U planes. The third element of array (pSrcBatchList[2]) represents a batch of V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.13 YUV422ToBGRBatch

Planar YUV422 to packed BGR batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

Functions

```
NppStatus nppiYUV422ToBGRBatch_8u_P3C3R_Ctx(const NppImageDescriptor *const  
                                              pSrcBatchList[3], NppImageDescriptor  
                                              *pDstBatchList, int nBatchSize, NppiSize  
                                              oSizeROI, NppStreamContext nppStreamCtx)
```

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed BGR batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of U planes. The third element of array (pSrcBatchList[2]) represents a batch of V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.

- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.14 YUV422ToBGRBatchAdvanced

Planar YUV422 to packed BGR batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

Functions

NppStatus **nppiYUV422ToBGRBatch_8u_P3C3R_Advanced_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed BGR batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of U planes. The third element of array (pSrcBatchList[2]) represents a batch of V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.15 RGBToYUV420

RGB to YUV420 color conversion.

Functions

NppStatus **nppiRGBToYUV420_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV420 color conversion. images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYUV420_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.16 YUV420ToRGB

YUV420 to RGB color conversion.

Functions

NppStatus **nppiYUV420ToRGB_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Planar-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV420ToRGB_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV420ToRGB_8u_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha (0xFF).

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYUV420ToRGB_8u_P3AC4R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.17 YUV420ToRGBBatch

Planar YUV420 to packed RGB batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

Functions

NppStatus **nppiYUV420ToRGBBatch_8u_P3C3R_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed RGB batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents Y planes. The second element of array (pSrcBatchList[1]) represents U planes. The third element of array (pSrcBatchList[2]) represents V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.18 YUV420ToRGBBatchAdvanced

Planar YUV420 to packed RGB batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

Functions

```
NppStatus nppiYUV420ToRGBBatch_8u_P3C3R_Advanced_Ctx(const NppImageDescriptor *const
    pSrcBatchList[3],
    NppImageDescriptor
    *pDstBatchList, int nBatchSize,
    NppiSize oMaxSizeROI,
    NppStreamContext
    nppStreamCtx)
```

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed RGB batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of U planes. The third element of array (pSrcBatchList[2]) represents a batch of V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.19 NV12ToRGB

NV12 to RGB color conversion.

Functions

NppStatus **nppiNV12ToRGB_8u_P2C3R_Ctx**(const *Npp8u* *const pSrc[2], int rSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV12 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for UV plane).
- ▶ **rSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each source plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiNV12ToRGB_709HDTV_8u_P2C3R_Ctx**(const *Npp8u* *const pSrc[2], int rSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV12 to 3 channel 8-bit unsigned packed RGB 709 HDTV full color conversion.

Note that HDTV conversion assumes full color range of 0 - 255, use CSC version for limited range color.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for UV plane).
- ▶ **rSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiNV12ToRGB_709CSC_8u_P2C3R_Ctx**(const *Npp8u* *const pSrc[2], int rSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV12 to 3 channel 8-bit unsigned packed RGB 709 CSC color conversion.

Note that HDTV conversion assumes full color range of 0 - 255, use CSC version for limited range color.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for UV plane).
- ▶ **rSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiNV12ToRGB_601_8u_P2C3R_Ctx**(const *Npp8u* *const pSrc[2], int rSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV12 to 3 channel 8-bit unsigned packed RGB BT.601 full-range color conversion (ITU-T T.871 / JFIF).

Coefficients derived from BT.601 luma weights (Kr=0.299, Kb=0.114) with full digital range Y=[0..255], Cb/Cr=[0..255] (offset 128). Use this variant for cameras that signal full-range BT.601 NV12 (e.g. OAK-D VPU encoder output), as distinct from the unspecified-range default nppiNV12ToRGB_8u_P2C3R and the BT.709 variants.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for UV plane).
- ▶ **rSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.20 NV21ToRGB

NV21 to RGB color conversion.

Functions

NppStatus **nppiNV21ToRGB_8u_P2C4R_Ctx**(const *Npp8u* *const pSrc[2], int rSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV21 to 4 channel 8-bit unsigned packed RGBA color conversion with constant alpha (0xFF).

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for VU plane).
- ▶ **rSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.21 BGRTToYUV420

BGR to YUV420 color conversion.

Functions

NppStatus **nppiBGRTToYUV420_8u_AC4P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YUV420 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes***1.8.3.22 YUV420ToBGR**

YUV420 to BGR color conversion.

Functions

NppStatus **nppiYUV420ToBGR_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiYUV420ToBGR_8u_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha (0xFF).

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

1.8.3.23 YUV420ToBGRBatch

Planar YUV420 to packed BGR batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

Functions

NppStatus **nppiYUV420ToBGRBatch_8u_P3C3R_Ctx**(const *NppImageDescriptor* *const
pSrcBatchList[3], *NppImageDescriptor*
*pDstBatchList, int nBatchSize, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed BGR batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents Y planes. The second element of array (pSrcBatchList[1]) represents U planes. The third element of array (pSrcBatchList[2]) represents V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.24 YUV420ToBGRBatchAdvanced

Planar YUV420 to packed BGR batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

Functions

NppStatus **nppiYUV420ToBGRBatch_8u_P3C3R_Advanced_Ctx**(const *NppImageDescriptor* *const
pSrcBatchList[3],
NppImageDescriptor
*pDstBatchList, int nBatchSize,
NppiSize oMaxSizeROI,
NppStreamContext
nppStreamCtx)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed BGR batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of U planes. The third element of array (pSrcBatchList[2]) represents a batch of V planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppiImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.25 NV12ToBGR

NV12 to BGR color conversion.

Functions

NppStatus **nppiNV12ToBGR_8u_P2C3R_Ctx**(const *Npp8u* *const pSrc[2], int rSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV12 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for UV plane).
- ▶ **rSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiNV12ToBGR_709HDTV_8u_P2C3R_Ctx**(const *Npp8u* *const pSrc[2], int rSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV12 to 3 channel 8-bit unsigned packed RGB 709 HDTV full color conversion.

Note that HDTV conversion assumes full color range of 0 - 255, use CSC version for limited range color.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for UV plane).
- ▶ **rSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiNV12ToBGR_709CSC_8u_P2C3R_Ctx**(const *Npp8u* *const pSrc[2], int rSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV12 to 3 channel 8-bit unsigned packed RGB 709 CSC color conversion.

Note that HDTV conversion assumes full color range of 0 - 255, use CSC version for limited range color.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for UV plane).
- ▶ **rSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiNV12ToBGR_601_8u_P2C3R_Ctx**(const *Npp8u* *const pSrc[2], int rSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV12 to 3 channel 8-bit unsigned packed BGR BT.601 full-range color conversion (ITU-T T.871 / JFIF).

Coefficients derived from BT.601 luma weights (Kr=0.299, Kb=0.114) with full digital range Y=[0..255], Cb/Cr=[0..255] (offset 128).

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for UV plane).
- ▶ **rSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.26 NV21ToBGR

NV21 to BGR color conversion.

Functions

NppStatus **nppiNV21ToBGR_8u_P2C4R_Ctx**(const *Npp8u* *const pSrc[2], int rSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned planar NV21 to 4 channel 8-bit unsigned packed BGRA color conversion with constant alpha (0xFF).

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array* (one for Y plane, one for VU plane).
- ▶ **rSrcStep** – *Source-Planar-Image Line Step*. Same value is used for each plane.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.27 RGBToYCbCr

RGB to YCbCr color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YCbCr. In the YCbCr model, Y is defined to have a nominal range [16..235], while Cb and Cr are defined to have a range [16..240], with the value of 128 as corresponding to zero.

```
Npp32f nY = 0.257F * R + 0.504F * G + 0.098F * B + 16.0F;  
Npp32f nCb = -0.148F * R - 0.291F * G + 0.439F * B + 128.0F;  
Npp32f nCr = 0.439F * R - 0.368F * G - 0.071F * B + 128.0F;
```

Functions

NppStatus **nppiRGBToYCbCr_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit packed YCbCr color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYCbCr_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel unsigned 8-bit packed YCbCr with alpha color conversion, not affecting alpha.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYCbCr_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel planar 8-bit unsigned RGB to 3 channel planar 8-bit YCbCr color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYCbCr_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit planar YCbCr color conversion. images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYCbCr_8u_AC4P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.28 YCbCrToRGB

YCbCr to RGB color conversion.

Here is how NPP converts YCbCr to gamma corrected RGB or BGR. The output RGB values are saturated to the range [0..255].

```
Npp32f nY = 1.164F * ((Npp32f)Y - 16.0F);  
Npp32f nR = ((Npp32f)Cr - 128.0F); Npp32f nB = ((Npp32f)Cb  
- 128.0F); Npp32f nG = nY - 0.813F * nR - 0.392F * nB; if (nG > 255.0F)  
    nG = 255.0F;  
nR = nY + 1.596F * nR;  
if (nR > 255.0F)  
    nR = 255.0F;  
nB = nY + 2.017F * nB;  
if (nB > 255.0F)  
    nB = 255.0F;
```

Functions

NppStatus **nppiYCbCrToRGB_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCrToRGB_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed YCbCr with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion, not affecting alpha.

Alpha channel is the last channel and is not processed.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.

- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCrToRGB_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Planar-Image Pointer Array.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCrToRGB_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCrToRGB_8u_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nAval, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAval** – 8-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.29 YCbCrToRGBBatch

YCbCr to RGB batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

NPP converts YCbCr to gamma corrected RGB the same way as in *YCbCr To RGB*.

Functions

NppStatus **nppiYCbCrToRGBBatch_8u_C3R_Ctx**(const *NppImageDescriptor* *pSrcBatchList,
NppImageDescriptor *pDstBatchList, int
nBatchSize, *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed RGB batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input and output images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer.*
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer.*
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCrToRGBBatch_8u_P3C3R_Ctx**(const *NppImageDescriptor* *const
pSrcBatchList[3], *NppImageDescriptor*
*pDstBatchList, int nBatchSize, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed RGB batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pDstBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.30 YCbCrToRGBBatchAdvanced

YCbCr to RGB batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

NPP converts YCbCr to gamma corrected RGB the same way as in *YCbCr To RGB*.

Functions

```
NppStatus nppiYCbCrToRGBBatch_8u_C3R_Advanced_Ctx( const NppImageDescriptor
                                                    *pSrcBatchList, NppImageDescriptor
                                                    *pDstBatchList, int nBatchSize,
                                                    NppiSize oMaxSizeROI,
                                                    NppStreamContext nppStreamCtx)
```

3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed RGB batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer*.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCrToRGBBatch_8u_P3C3R_Advanced_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3],
NppImageDescriptor *pDstBatchList, int nBatchSize,
NppiSize oMaxSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed RGB batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- **pDstBatchList** – *Destination-Batch-Images Pointer.*
- **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.31 YCbCrToBGR

YCbCr to BGR color conversion.

Functions

NppStatus **nppiYCbCrToBGR_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*

- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCrToBGR_8u_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nAval, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nAval** – 8-bit unsigned alpha constant.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.32 YCbCrToBGRBatch

YCbCr to BGR batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

NPP converts YCbCr to gamma corrected BGR the same way as in *YCbCrToBGR*.

Functions

NppStatus **nppiYCbCrToBGRBatch_8u_C3R_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed BGR batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input and output images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer*.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCrToBGRBatch_8u_P3C3R_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.33 YCbCrToBGRBatchAdvanced

YCbCr to BGR batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

NPP converts YCbCr to gamma corrected BGR the same way as in *YCbCrToBGR*.

Functions

NppStatus **nppiYCbCrToBGRBatch_8u_C3R_Advanced_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed BGR batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer*.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCrToBGRBatch_8u_P3C3R_Advanced_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.34 YCbCrToBGR709CSC

YCbCr to BGR_709CSC color conversion.

Functions

NppStatus **nppiYCbCrToBGR_709CSC_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR_709CSC color conversion.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCrToBGR_709CSC_8u_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nAval, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR_709CSC color conversion with constant alpha.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nAval** – 8-bit unsigned alpha constant.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes***1.8.3.35 RGBToYCbCr422**

RGB to YCbCr422 color conversion.

Functions

NppStatus **nppiRGBToYCbCr422_8u_C3C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiRGBToYCbCr422_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr422 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiRGBToYCbCr422_8u_P3C2R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.36 YCbCr422ToRGB

YCbCr422 to RGB color conversion.

Functions

NppStatus **nppiYCbCr422ToRGB_8u_C2C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToRGB_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3],
int nDstStep, *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar RGB color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToRGB_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst, int nDstStep, *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array*.
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.37 YCbCr422ToRGBBatch

Planar YCbCr422 to packed RGB batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

Functions

NppStatus **nppiYCbCr422ToRGBBatch_8u_P3C3R_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed RGB batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.38 YCbCr422ToRGBBatchAdvanced

Planar YCbCr422 to packed RGB batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

Functions

NppStatus **nppiYCbCr422ToRGBBatch_8u_P3C3R_Advanced_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed RGB batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.39 RGBToYCrCb422

RGB to YCrCb422 color conversion.

Functions

NppStatus **nppiRGBToYCrCb422_8u_C3C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYCrCb422_8u_P3C2R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.40 YCrCb422ToRGB

YCrCb422 to RGB color conversion.

Functions

NppStatus **nppiYCrCb422ToRGB_8u_C2C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed RGB color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCrCb422ToRGB_8u_C2P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar RGB color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.41 YCbCr422ToBGR

YCbCr422 to BGR color conversion.

Functions

NppStatus **nppiYCbCr422ToBGR_8u_C2C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed BGR color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToBGR_8u_C2C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nAval, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed YCrCb422 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nAval** – 8-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToBGR_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst, int nDstStep, *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed BGR color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array*.
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.42 YCbCr422ToBGRBatch

Planar YCbCr422 to packed BGR batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

Functions

NppStatus **nppiYCbCr422ToBGRBatch_8u_P3C3R_Ctx**(const *NppiImageDescriptor* *const
pSrcBatchList[3], *NppiImageDescriptor*
*pDstBatchList, int nBatchSize, *NppiSize*
oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed BGR batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pDstBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.43 YCbCr422ToBGRBatchAdvanced

Planar YCbCr422 to packed BGR batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

Functions

```
NppStatus nppiYCbCr422ToBGRBatch_8u_P3C3R_Advanced_Ctx( const NppImageDescriptor
                                                         *const pSrcBatchList[3],
                                                         NppImageDescriptor
                                                         *pDstBatchList, int nBatchSize,
                                                         NppiSize oMaxSizeROI,
                                                         NppStreamContext
                                                         nppStreamCtx )
```

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed BGR batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.

- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.44 RGBToCbYCr422

RGB to CbYCr422 color conversion.

Functions

NppStatus **nppiRGBToCbYCr422_8u_C3C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed CbYCr422 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToCbYCr422Gamma_8u_C3C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB first gets forward gamma corrected then converted to 2 channel 8-bit unsigned packed CbYCr422 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.45 CbYCr422ToRGB

CbYCr422 to RGB color conversion.

Functions

NppStatus **nppiCbYCr422ToRGB_8u_C2C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed CbYCrC22 to 3 channel 8-bit unsigned packed RGB color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.46 BGRTCbYCr422

BGR to CbYCr422 color conversion.

Functions

NppStatus **nppiBGRTCbYCr422_8u_AC4C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.47 BGRTToCbYCr422_709HDTV

BGR to CbYCr422_709HDTV color conversion.

Functions

NppStatus **nppiBGRTToCbYCr422_709HDTV_8u_C3C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed CbYCr422_709HDTV color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRTToCbYCr422_709HDTV_8u_AC4C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422_709HDTV color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.48 CbYCr422ToBGR

CbYCr422 to BGR color conversion.

Functions

NppStatus **nppiCbYCr422ToBGR_8u_C2C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nAval, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR color conversion with alpha.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAval** – 8-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.49 CbYCr422ToBGR 709HDTV

CbYCr422 to BGR_709HDTV color conversion.

Functions

NppStatus **nppiCbYCr422ToBGR_709HDTV_8u_C2C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned packed BGR_709HDTV color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCbYCr422ToBGR_709HDTV_8u_C2C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nAval, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR_709HDTV color conversion with constant alpha.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAval** – 8-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.50 RGBToYCbCr420

RGB to YCbCr420 color conversion.

Functions

NppStatus **nppiRGBToYCbCr420_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3],
int rDstStep[3], *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.51 YCbCr420ToRGB

YCbCr420 to RGB color conversion.

Functions

NppStatus **nppiYCbCr420ToRGB_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst, int nDstStep, *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array*.
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.52 YCbCr420ToRGBBatch

Planar YCbCr420 to packed RGB batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

Functions

NppStatus **nppiYCbCr420ToRGBBatch_8u_P3C3R_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed RGB batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.53 YCbCr420ToRGBBatchAdvanced

Planar YCbCr420 to packed RGB batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

Functions

NppStatus **nppiYCbCr420ToRGBBatch_8u_P3C3R_Advanced_Ctx**(const *NppImageDescriptor* *const pSrcBatchList[3], *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed RGB batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- **pDstBatchList** – *Destination-Batch-Images Pointer*.
- **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.54 RGBToYCrCb420

RGB to YCrCb420 color conversion.

Functions

NppStatus **nppiRGBToYCrCb420_8u_AC4P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

images.

Parameters

- **pSrc** – *Source-Image Pointer*.

- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.55 YCrCb420ToRGB

YCrCb420 to RGB color conversion.

Functions

NppStatus **nppiYCrCb420ToRGB_8u_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst, int nDstStep, *NppiSize* oSizeROI,
Npp8u nAval, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCrCb420 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array*.
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nAval** – 8-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.56 BGRToYCbCr420

BGR to YCbCr420 color conversion.

Functions

NppStatus **nppiBGRToYCbCr420_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3],
int rDstStep[3], *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr420_8u_AC4P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.57 BGRTToYCbCr420_709CSC

BGR to YCbCr420_709CSC color conversion.

Functions

NppStatus **nppiBGRTToYCbCr420_709CSC_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420_709CSC color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRTToYCbCr420_709CSC_8u_AC4P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420_709CSC color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.58 BGRToYCbCr420_709HDTV

BGR to YCbCr420_709HDTV color conversion.

Functions

NppStatus **nppiBGRToYCbCr420_709HDTV_8u_AC4P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420_709HDTV color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.59 BGRToYCrCb420_709CSC

BGR to YCrCb420_709CSC color conversion.

Functions

NppStatus **nppiBGRToYCrCb420_709CSC_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420_709CSC color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCrCb420_709CSC_8u_AC4P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420_709CSC color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.60 YCbCr420ToBGR

YCbCr420 to BGR color conversion.

Functions

NppStatus **nppiYCbCr420ToBGR_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiYCbCr420ToBGR_8u_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst, int nDstStep, *NppiSize* oSizeROI,
Npp8u nAval, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array*.
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nAval** – 8-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes***1.8.3.61 YCbCr420ToBGRBatch**

Planar YCbCr420 to packed BGR batch color conversion with a single *Region-Of-Interest (ROI)* for all pairs of input/output images provided in batches.

Functions

NppStatus **nppiYCbCr420ToBGRBatch_8u_P3C3R_Ctx**(const *NppImageDescriptor* *const
pSrcBatchList[3], *NppImageDescriptor*
*pDstBatchList, int nBatchSize, *NppiSize*
oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR batch color conversion for a single ROI.

Provided oSizeROI will be used for all pairs of input planes making input images and output packed images passed in pSrcBatchList and pSrcBatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.

- ▶ **nBatchSize** – A number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.62 YCbCr420ToBGRBatchAdvanced

Planar YCbCr420 to packed BGR batch color conversion where each pair of input/output images from provided batches has own *Region-Of-Interest (ROI)*.

Functions

```
NppStatus nppiYCbCr420ToBGRBatch_8u_P3C3R_Advanced_Ctx( const NppImageDescriptor
                                                         *const pSrcBatchList[3],
                                                         NppImageDescriptor
                                                         *pDstBatchList, int nBatchSize,
                                                         NppiSize oMaxSizeROI,
                                                         NppStreamContext
                                                         nppStreamCtx )
```

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR batch color conversion where each pair of input/output images has own ROI.

Provided oMaxSizeROI must contain the maximum width and the maximum height of all ROIs defined in pDstBatchList. API user must ensure that ROI from pDstBatchList for each pair of input and output images does not go beyond the borders of images in each pair.

Parameters

- ▶ **pSrcBatchList** – An array where each element is a batch of images representing one of planes in planar images, *Source-Batch-Images Pointer*. The first element of array (pSrcBatchList[0]) represents a batch of Y planes. The second element of array (pSrcBatchList[1]) represents a batch of Cb planes. The third element of array (pSrcBatchList[2]) represents a batch of Cr planes.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer*.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)*, must contain the maximum width and the maximum height from all destination ROIs used for processing data.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.63 YCbCr420ToBGR 709CSC

YCbCr420_709CSC to BGR color conversion.

Functions

NppStatus **nppiYCbCr420ToBGR_709CSC_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR_709CSC color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.64 YCbCr420ToBGR 709HDTV

YCbCr420_709HDTV to BGR color conversion.

Functions

NppStatus **nppiYCbCr420ToBGR_709HDTV_8u_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp8u* nAval, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR_709HDTV color conversion with constant alpha.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAval** – 8-bit unsigned alpha constant.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.65 BGRToYCrCb420

BGR to YCrCb420 color conversion.

Functions

NppStatus **nppiBGRToYCrCb420_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3],
int rDstStep[3], *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

images.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Planar-Image Pointer Array.*
- **rDstStep** – Destination-Planar-Image Line Step Array.
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCrCb420_8u_AC4P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

images.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Planar-Image Pointer Array.*
- **rDstStep** – Destination-Planar-Image Line Step Array.
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes***1.8.3.66 BGRToYCbCr411**

BGR to YCbCr411 color conversion.

Functions

NppStatus **nppiBGRToYCbCr411_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3],
int rDstStep[3], *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiBGRToYCbCr411_8u_AC4P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int rDstStep[3], *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **rDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

1.8.3.67 BGRToYCbCr

BGR to YCbCr color conversion.

Functions

NppStatus **nppiBGRToYCbCr_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr color conversion. images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr_8u_AC4P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – Destination-Planar-Image Line Step Array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr_8u_AC4P4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[4], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar YCbCr color conversion.

images.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Planar-Image Line Step Array.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.68 YCbCr411ToBGR

YCbCr411 to BGR color conversion.

Functions

NppStatus **nppiYCbCr411ToBGR_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst, int nDstStep, *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToBGR_8u_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst, int nDstStep, *NppiSize* oSizeROI,
Npp8u nAval, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAval** – 8-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.69 YCbCr411ToRGB

YCbCr411 to RGB color conversion.

Functions

NppStatus **nppiYCbCr411ToRGB_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst, int nDstStep, *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToRGB_8u_P3C4R_Ctx**(const *Npp8u* *const pSrc[3], int rSrcStep[3],
Npp8u *pDst, int nDstStep, *NppiSize* oSizeROI,
Npp8u nAval, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*

- ▶ **rSrcStep** – *Source-Planar-Image Line Step Array.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nAval** – 8-bit unsigned alpha constant.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.70 RGBToXYZ

RGB to XYZ color conversion.

Here is how NPP converts gamma corrected RGB or BGR to XYZ.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F *
    ↪ nNormalizedB;
if (nX > 1.0F)
    nX = 1.0F;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F *
    ↪ nNormalizedB;
if (nY > 1.0F)
    nY = 1.0F;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F *
    ↪ nNormalizedB;
if (nZ > 1.0F)
    nZ = 1.0F;
X = (Npp8u)(nX * 255.0F);
Y = (Npp8u)(nY * 255.0F);
Z = (Npp8u)(nZ * 255.0F);
```

Functions

NppStatus **nppiRGBToXYZ_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed XYZ color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToXYZ_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed XYZ with alpha color conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.71 XYZToRGB

XYZ to RGB color conversion.

Here is how NPP converts XYZ to gamma corrected RGB or BGR. The code assumes that X,Y, and Z values are in the range [0..1].

```
Npp32f nNormalizedX = (Npp32f)X * 0.003921569F; // / 255.0F
Npp32f nNormalizedY = (Npp32f)Y * 0.003921569F;
Npp32f nNormalizedZ = (Npp32f)Z * 0.003921569F;
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F *
↪ nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F *
↪ nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;
Npp32f nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F *
↪ nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);
```


Functions

NppStatus **nppiXYZToRGB_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed XYZ to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiXYZToRGB_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed XYZ with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.72 RGBToLUV

RGB to LUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to CIE LUV using the CIE XYZ D65 white point with a Y luminance of 1.0. The computed values of the L component are in the range [0..100], U component in the range [-134..220], and V component in the range [-140..122]. The code uses cbrtf() the 32 bit floating point cube root math function.

```

// use CIE D65 chromaticity coordinates
#define nCIE_XYZ_D65_xn 0.312713F
#define nCIE_XYZ_D65_yn 0.329016F
#define nn_DIVISOR (-2.0F * nCIE_XYZ_D65_xn + 12.0F * nCIE_XYZ_D65_yn + 3.0F)
#define nun (4.0F * nCIE_XYZ_D65_xn / nn_DIVISOR)
#define nvN (9.0F * nCIE_XYZ_D65_yn / nn_DIVISOR)

// First convert to XYZ
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F *
↪ nNormalizedB;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F *
↪ nNormalizedB;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F *
↪ nNormalizedB;
// Now calculate LUV from the XYZ value
Npp32f nTemp = nX + 15.0F * nY + 3.0F * nZ;
Npp32f nu = 4.0F * nX / nTemp;
Npp32f nv = 9.0F * nY / nTemp;
Npp32f nL = 116.0F * cbrtf(nY) - 16.0F;
if (nL < 0.0F)
    nL = 0.0F;
if (nL > 100.0F)
    nL = 100.0F;
nTemp = 13.0F * nL;
Npp32f nU = nTemp * (nu - nun);
if (nU < -134.0F)
    nU = -134.0F;
if (nU > 220.0F)
    nU = 220.0F;
Npp32f nV = nTemp * (nv - nvN);
if (nV < -140.0F)
    nV = -140.0F;
if (nV > 122.0F)
    nV = 122.0F;
L = (Npp8u)(nL * 255.0F * 0.01F); // / 100.0F
U = (Npp8u)((nU + 134.0F) * 255.0F * 0.0028249F); // / 354.0F
V = (Npp8u)((nV + 140.0F) * 255.0F * 0.0038168F); // / 262.0F

```

Functions

NppStatus **nppiRGBToLUV_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed LUV color conversion.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **pDst** – *Destination-Image Pointer*.

- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToLUV_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed LUV with alpha color conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.73 LUVToRGB

LUV to RGB color conversion.

Here is how NPP converts CIE LUV to gamma corrected RGB or BGR using the CIE XYZ D65 white point with a Y luminance of 1.0. The code uses powf() the 32 bit floating point power math function.

```
// use CIE D65 chromaticity coordinates
#define nCIE_XYZ_D65_xn 0.312713F
#define nCIE_XYZ_D65_yn 0.329016F
#define nn_DIVISOR (-2.0F * nCIE_XYZ_D65_xn + 12.0F * nCIE_XYZ_D65_yn + 3.0F)
#define nun (4.0F * nCIE_XYZ_D65_xn / nn_DIVISOR)
#define nvn (9.0F * nCIE_XYZ_D65_yn / nn_DIVISOR)

// First convert normalized LUV back to original CIE LUV range
Npp32f nL = (Npp32f)L * 100.0F * 0.003921569F; // / 255.0F
Npp32f nU = ((Npp32f)U * 354.0F * 0.003921569F) - 134.0F;
Npp32f nV = ((Npp32f)V * 262.0F * 0.003921569F) - 140.0F;
// Now convert LUV to CIE XYZ
Npp32f nTemp = 13.0F * nL;
Npp32f nu = nU / nTemp + nun;
Npp32f nv = nV / nTemp + nvn;
Npp32f nNormalizedY;
if (nL > 7.9996248F)
{
    nNormalizedY = (nL + 16.0F) * 0.008621F; // / 116.0F
```

(continues on next page)

(continued from previous page)

```

    nNormalizedY = powf(nNormalizedY, 3.0F);
}
else
{
    nNormalizedY = nL * 0.001107F; // / 903.3F
}
Npp32f nNormalizedX = (-9.0F * nNormalizedY * nu) / ((nu - 4.0F) * nv - nu * nv);
Npp32f nNormalizedZ = (9.0F * nNormalizedY - 15.0F * nv * nNormalizedY - nv *
    ↪ nNormalizedX) / (3.0F * nv);
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F *
    ↪ nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
if (nR < 0.0F)
    nR = 0.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F *
    ↪ nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;
if (nG < 0.0F)
    nG = 0.0F;
Npp32f nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F *
    ↪ nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
if (nB < 0.0F)
    nB = 0.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);

```

Functions

NppStatus **nppiLUVToRGB_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed LUV to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiLUVToRGB_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed LUV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.74 BGRTToLab

BGR to Lab color conversion.

This is how NPP converts gamma corrected BGR or RGB to Lab using the CIE Lab D65 white point with a Y luminance of 1.0. The computed values of the L component are in the range [0..100], a and b component values are in the range [-128..127]. The code uses cbrtf() the 32 bit floating point cube root math function.

```
// use CIE Lab chromaticity coordinates
#define nCIE_LAB_D65_xn 0.950455F
#define nCIE_LAB_D65_yn 1.0F
#define nCIE_LAB_D65_zn 1.088753F
// First convert to XYZ
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F *
↪ nNormalizedB;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F *
↪ nNormalizedB;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F *
↪ nNormalizedB;
Npp32f nL = cbrtf(nY);
Npp32f nA; Npp32f nB; Npp32f nfX = nX * 1.052128F; // / nCIE_LAB_D65_xn; Npp32f nfY =
↪ nY; Npp32f nfZ = nZ * 0.918482F; // /
nCIE_LAB_D65_zn; nfY = nL - 16.0F; nL = 116.0F * nL - 16.0F; nA = cbrtf(nfX) - 16.0F;
↪ nA = 500.0F
* (nA - nfY); nB = cbrtf(nfZ) - 16.0F; nB = 200.0F * (nfY - nB); // Now scale Lab
↪ range nL = nL * 255.0F
* 0.01F; // / 100.0F nA = nA + 128.0F; nB = nB + 128.0F; L = (Npp8u)nL; a = (Npp8u)nA;
↪ b = (Npp8u)nB;
```

Functions

NppStatus **nppiBGRToLab_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed Lab color conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.75 LabToBGR

Lab to BGR color conversion.

This is how NPP converts Lab to gamma corrected BGR or RGB using the CIE Lab D65 white point with a Y luminance of 1.0. The code uses powf() the 32 bit floating point power math function.

```
// use CIE Lab chromaticity coordinates
#define nCIE_LAB_D65_xn 0.950455F
#define nCIE_LAB_D65_yn 1.0F
#define nCIE_LAB_D65_zn 1.088753F
// First convert Lab back to original range then to CIE XYZ
Npp32f nL = (Npp32f)L * 100.0F * 0.003921569F; // / 255.0F
Npp32f nA = (Npp32f)a - 128.0F;
Npp32f nB = (Npp32f)b - 128.0F;
Npp32f nP = (nL + 16.0F) * 0.008621F; // / 116.0F
Npp32f nNormalizedY = nP * nP * nP; // powf(nP, 3.0F);
Npp32f nNormalizedX = nCIE_LAB_D65_xn * powf((nP + nA * 0.002F), 3.0F); // / 500.0F
Npp32f nNormalizedZ = nCIE_LAB_D65_zn * powf((nP - nB * 0.005F), 3.0F); // / 200.0F
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F *
↪ nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F *
↪ nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;
nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F * nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);
```

Functions

NppStatus **nppiLabToBGR_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed Lab to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.76 RGBToYCC

RGB to PhotoYCC color conversion.

This is how NPP converts gamma corrected BGR or RGB to PhotoYCC. The computed Y, C1, C2 values are then quantized and converted to fit in the range [0..1] before expanding to 8 bits.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nY = 0.299F * nNormalizedR + 0.587F * nNormalizedG + 0.114F * nNormalizedB;
Npp32f nC1 = nNormalizedB - nY;
nC1 = 111.4F * 0.003921569F * nC1 + 156.0F * 0.003921569F;
Npp32f nC2 = nNormalizedR - nY;
nC2 = 135.64F * 0.003921569F * nC2 + 137.0F * 0.003921569F;
nY = 1.0F * 0.713267F * nY; // / 1.402F
Y = (Npp8u)(nY * 255.0F);
C1 = (Npp8u)(nC1 * 255.0F);
C2 = (Npp8u)(nC2 * 255.0F);
```

Functions

NppStatus **nppiRGBToYCC_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YCC color conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*

- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYCC_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YCC with alpha color conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.77 YCCToRGB

PhotoYCC to RGB color conversion.

This is how NPP converts PhotoYCC to gamma corrected RGB or BGR.

```
Npp32f nNormalizedY = ((Npp32f)Y * 0.003921569F) * 1.3584F; // / 255.0F
Npp32f nNormalizedC1 = (((Npp32f)C1 * 0.003921569F) - 156.0F * 0.003921569F) * 2.
↪ 2179F;
Npp32f nNormalizedC2 = (((Npp32f)C2 * 0.003921569F) - 137.0F * 0.003921569F) * 1.
↪ 8215F;
Npp32f nR = nNormalizedY + nNormalizedC2;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = nNormalizedY - 0.194F * nNormalizedC1 - 0.509F * nNormalizedC2;
if (nG > 1.0F)
    nG = 1.0F;
Npp32f nB = nNormalizedY + nNormalizedC1;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);
```


Functions

NppStatus **nppiYCCToRGB_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed YCC to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCCToRGB_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed YCC with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.78 YCCKToCMYK_JPEG

This function partially converts JPEG YCCK to CMYK.

This is how NPP converts JPEG YCCK to CMYK. NPP only performs an initial YCC to RGB conversion using the 601 conversion coefficients and the RGB to CMY inversion leaving K unmodified. To complete this conversion to useful RGB values an additional RGB conversion needs to follow this function using the color profile contained in the YCCK JPEG file metadata section. NPP does not directly support this conversion but potentially `nppiColorTwist` can be used to perform it once the conversion coefficients are known.

```

Npp32f nY = static_cast<Npp32f>(Y);
Npp32f nC1 = static_cast<Npp32f>(Cb);
Npp32f nC2 = static_cast<Npp32f>(Cr);
Npp32f nR = nY + 1.402F * nC2 - 179.456F;
Npp32f nG = nY - 0.34414F * nC1 - 0.71414F * nC2 + 135.45984F;
Npp32f nB = nY + 1.772F * nC1 - 226.816F;

Npp8u nC = static_cast<Npp8u>(255.0F - nR);
Npp8u nM = static_cast<Npp8u>(255.0F - nG);
Npp8u nM = static_cast<Npp8u>(255.0F - nB);
Npp8u nK = K;

```

Functions

NppStatus **nppiYCKKToCMYK_JPEG_601_8u_P4R_Ctx**(const *Npp8u* *pSrc[4], int nSrcStep, *Npp8u* *pDst[4], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned planar JPEG YCKK color format to 4 channel 8-bit unsigned planar CMYK color conversion using 601 RGB color coefficients and CMY inversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Planar-Image Pointer Array*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.79 CMYKorYCKKJPEGToRGB

These functions convert JPEG CMYK or YCKK color format images to either planar or packed RGB images for images which need no color profile conversion.

Functions

NppStatus **nppiCMYKorYCKKToRGB_JPEG_8u_P4P3R_Ctx**(const *Npp8u* *pSrc[4], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned planar JPEG CMYK or YCKK color model image to 3 channel 8-bit unsigned planar RGB color model image without color profile conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiCMYK0rYCCKToRGB_JPEG_8u_P4C3R_Ctx**(const *Npp8u* *pSrc[4], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned planar JPEG CMYK or YCKK color model image to 3 channel 8-bit unsigned packed RGB color model image without color profile conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.80 YCKKJPEG0rCMYKToBGR

These functions convert JPEG CMYK or YCKK color format images to either planar or packed BGR images for images which need no color profile conversion.

Functions

NppStatus **nppiCMYK0rYCCKToBGR_JPEG_8u_P4P3R_Ctx**(const *Npp8u* *pSrc[4], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned planar JPEG CMYK or YCKK color model image to 3 channel 8-bit unsigned planar BGR color model image without color profile conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiCMYKtoYCKKtoBGR_JPEG_8u_P4C3R_Ctx**(const *Npp8u* *pSrc[4], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned planar JPEG CMYK or YCKK color model image to 3 channel 8-bit unsigned packed BGR color model image without color profile conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes***1.8.3.81 RGBtoHLS**

RGB to HLS color conversion.

This is how NPP converts gamma corrected RGB or BGR to HLS. This code uses the fmaxf() and fminf() 32 bit floating point math functions.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nS;
Npp32f nH;
// Lightness
Npp32f nMax = fmaxf(nNormalizedR, nNormalizedG);
nMax = fmaxf(nMax, nNormalizedB);
Npp32f nMin = fminf(nNormalizedR, nNormalizedG);
nMin = fminf(nMin, nNormalizedB);
Npp32f nL = (nMax + nMin) * 0.5F;
Npp32f nDivisor = nMax - nMin;
// Saturation
if (nDivisor == 0.0F) // achromatics case
{
    nS = 0.0F;
```

(continues on next page)

(continued from previous page)

```

    nH = 0.0F;
}
else // chromatics case
{
    if (nL > 0.5F)
        nS = nDivisor / (1.0F - (nMax + nMin - 1.0F));
    else
        nS = nDivisor / (nMax + nMin);
}
// Hue
Npp32f nCr = (nMax - nNormalizedR) / nDivisor;
Npp32f nCg = (nMax - nNormalizedG) / nDivisor;
Npp32f nCb = (nMax - nNormalizedB) / nDivisor;
if (nNormalizedR == nMax)
    nH = nCb - nCg;
else if (nNormalizedG == nMax)
    nH = 2.0F + nCr - nCb;
else if (nNormalizedB == nMax)
    nH = 4.0F + nCg - nCr;
nH = nH * 0.166667F; // / 6.0F
if (nH < 0.0F)
    nH = nH + 1.0F;
H = (Npp8u)(nH * 255.0F);
L = (Npp8u)(nL * 255.0F);
S = (Npp8u)(nS * 255.0F);

```

Functions

NppStatus **nppiRGBToHLS_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HLS color conversion.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToHLS_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.82 HLSToRGB

HLS to RGB color conversion.

This is how NPP converts HLS to gamma corrected RGB or BGR.

```

Npp32f nNormalizedH = (Npp32f)H * 0.003921569F; // / 255.0F
Npp32f nNormalizedL = (Npp32f)L * 0.003921569F;
Npp32f nNormalizedS = (Npp32f)S * 0.003921569F;
Npp32f nM1;
Npp32f nM2;
Npp32f nR;
Npp32f nG;
Npp32f nB;
Npp32f nh = 0.0F;
if (nNormalizedL <= 0.5F)
    nM2 = nNormalizedL * (1.0F + nNormalizedS);
else
    nM2 = nNormalizedL + nNormalizedS - nNormalizedL * nNormalizedS;
nM1 = 2.0F * nNormalizedL - nM2;
if (nNormalizedS == 0.0F)
    nR = nG = nB = nNormalizedL;
else
{
    nh = nNormalizedH + 0.3333F;
    if (nh > 1.0F)
        nh -= 1.0F;
}
Npp32f nMDiff = nM2 - nM1;
if (0.6667F < nh)
    nR = nM1;
else
{
    if (nh < 0.1667F)
        nR = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nR = nM2;
    else
        nR = nM1 + nMDiff * ( 0.6667F - nh ) * 6.0F; // / 0.1667F
}

```

(continues on next page)

(continued from previous page)

```

if (nR > 1.0F)
    nR = 1.0F;
nh = nNormalizedH;
if (0.6667F < nh)
    nG = nM1;
else
{
    if (nh < 0.1667F)
        nG = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nG = nM2;
    else
        nG = nM1 + nMDiff * (0.6667F - nh) * 6.0F; // / 0.1667F
}
if (nG > 1.0F)
    nG = 1.0F;
nh = nNormalizedH - 0.3333F;
if (nh < 0.0F)
    nh += 1.0F;
if (0.6667F < nh)
    nB = nM1;
else
{
    if (nh < 0.1667F)
        nB = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nB = nM2;
    else
        nB = nM1 + nMDiff * (0.6667F - nh) * 6.0F; // / 0.1667F
}
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);

```

Functions

NppStatus **nppiHLSToRGB_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiHLSToRGB_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.83 BGRTToHLS

BGR to HLS color conversion.

Functions

NppStatus **nppiBGRTToHLS_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToHLS_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar HLS color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToHLS_8u_AC4P4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[4], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToHLS_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned packed HLS color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRTToHLS_8u_AP4C4R_Ctx**(const *Npp8u* *const pSrc[4], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRTToHLS_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar HLS color conversion.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Planar-Image Pointer Array.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRTToHLS_8u_AP4R_Ctx**(const *Npp8u* *const pSrc[4], int nSrcStep, *Npp8u* *pDst[4], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.84 HLSToBGR

HLS to BGR color conversion.

Functions

NppStatus **nppiHLSToBGR_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned planar BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiHLSToBGR_8u_AC4P4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[4], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiHLSToBGR_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3],
int nDstStep, *NppiSize* oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned planar BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiHLSToBGR_8u_AP4R_Ctx**(const *Npp8u* *const pSrc[4], int nSrcStep, *Npp8u*
*pDst[4], int nDstStep, *NppiSize* oSizeROI,
NppStreamContext nppStreamCtx)

4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Planar-Image Pointer Array.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiHLSToBGR_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst,
int nDstStep, *NppiSize* oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Planar-Image Pointer Array.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiHLSToBGR_8u_AP4C4R_Ctx**(const *Npp8u* *const pSrc[4], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.

Parameters

- **pSrc** – *Source-Planar-Image Pointer Array.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.85 RBGToHSV

RGB to HSV color conversion.

This is how NPP converts gamma corrected RGB or BGR to HSV. This code uses the fmaxf() and fminf() 32 bit floating point math functions.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nS;
Npp32f nH;
// Value
Npp32f nV = fmaxf(nNormalizedR, nNormalizedG);
nV = fmaxf(nV, nNormalizedB);
// Saturation
Npp32f nTemp = fminf(nNormalizedR, nNormalizedG);
nTemp = fminf(nTemp, nNormalizedB);
Npp32f nDivisor = nV - nTemp;
if (nV == 0.0F) // achromatics case
{
    nS = 0.0F;
    nH = 0.0F;
}
else // chromatics case
    nS = nDivisor / nV;
// Hue:
```

(continues on next page)

(continued from previous page)

```

Npp32f nCr = (nV - nNormalizedR) / nDivisor;
Npp32f nCg = (nV - nNormalizedG) / nDivisor;
Npp32f nCb = (nV - nNormalizedB) / nDivisor;
if (nNormalizedR == nV)
    nH = nCb - nCg;
else if (nNormalizedG == nV)
    nH = 2.0F + nCr - nCb;
else if (nNormalizedB == nV)
    nH = 4.0F + nCg - nCr;
nH = nH * 0.166667F; // / 6.0F
if (nH < 0.0F)
    nH = nH + 1.0F;
H = (Npp8u)(nH * 255.0F);
S = (Npp8u)(nS * 255.0F);
V = (Npp8u)(nV * 255.0F);

```

Functions

NppStatus **nppiRGBToHSV_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HSV color conversion.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToHSV_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HSV with alpha color conversion.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.

► **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.86 HSVToRGB

HSV to RGB color conversion.

This is how NPP converts HSV to gamma corrected RGB or BGR. This code uses the floorf() 32 bit floating point math function.

```
Npp32f nNormalizedH = (Npp32f)H * 0.003921569F; // / 255.0F
Npp32f nNormalizedS = (Npp32f)S * 0.003921569F;
Npp32f nNormalizedV = (Npp32f)V * 0.003921569F;
Npp32f nR;
Npp32f nG;
Npp32f nB;
if (nNormalizedS == 0.0F)
{
    nR = nG = nB = nNormalizedV;
}
else
{
    if (nNormalizedH == 1.0F)
        nNormalizedH = 0.0F;
    else
        nNormalizedH = nNormalizedH * 6.0F; // / 0.1667F
}
Npp32f nI = floorf(nNormalizedH);
Npp32f nF = nNormalizedH - nI;
Npp32f nM = nNormalizedV * (1.0F - nNormalizedS);
Npp32f nN = nNormalizedV * (1.0F - nNormalizedS * nF);
Npp32f nK = nNormalizedV * (1.0F - nNormalizedS * (1.0F - nF));
if (nI == 0.0F)
    { nR = nNormalizedV; nG = nK; nB = nM; }
else if (nI == 1.0F)
    { nR = nN; nG = nNormalizedV; nB = nM; }
else if (nI == 2.0F)
    { nR = nM; nG = nNormalizedV; nB = nK; }
else if (nI == 3.0F)
    { nR = nM; nG = nN; nB = nNormalizedV; }
else if (nI == 4.0F)
    { nR = nK; nG = nM; nB = nNormalizedV; }
else if (nI == 5.0F)
    { nR = nNormalizedV; nG = nM; nB = nN; }
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);
```

Functions

NppStatus **nppiHSVToRGB_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed HSV to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiHSVToRGB_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned packed HSV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.8.3.87 JPEG Color Conversion

The set of JPEG color conversion functions available in the library.

RGBToYCbCr_JPEG Planar to planar.

JPEG RGB to YCbCr color conversion.

NppStatus **nppiRGBToYCbCr420_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYCbCr422_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YCbCr422 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYCbCr411_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*

- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYCbCr444_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YCbCr444 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr420_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr422_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar YCbCr422 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr411_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr444_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar YCbCr444 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr420ToRGB_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3],
Npp8u *pDst[3], int nDstStep, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToRGB_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3],
Npp8u *pDst[3], int nDstStep, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToRGB_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3],
Npp8u *pDst[3], int nDstStep, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr444ToRGB_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr444 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr420ToBGR_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToBGR_JPEG_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar BGR color conversion.

Parameters

RGBToYCbCr_JPEG Planar to packed.

JPEG RGB to YCbCr color conversion.

```
NppStatus nppiRGBToYCbCr420_JPEG_8u_C3P3R_Ctx( const Npp8u *pSrc, int nSrcStep, Npp8u
                                                *pDst[3], int aDstStep[3], NppiSize
                                                oSizeROI, NppStreamContext
                                                nppStreamCtx)
```

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiRGBToYCbCr422_JPEG_8u_C3P3R_Ctx( const Npp8u *pSrc, int nSrcStep, Npp8u
                                                *pDst[3], int aDstStep[3], NppiSize
                                                oSizeROI, NppStreamContext
                                                nppStreamCtx)
```

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr422 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiRGBToYCbCr411_JPEG_8u_C3P3R_Ctx( const Npp8u *pSrc, int nSrcStep, Npp8u
                                                *pDst[3], int aDstStep[3], NppiSize
                                                oSizeROI, NppStreamContext
                                                nppStreamCtx)
```

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRGBToYCbCr444_JPEG_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr444 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr420_JPEG_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr422_JPEG_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr422 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr411_JPEG_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int aDstStep[3], *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **aDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiBGRToYCbCr444_JPEG_8u_C3P3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr444 color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*

- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr420ToRGB_JPEG_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int
aSrcStep[3], *Npp8u* *pDst, int nDstStep,
NppiSize oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned packed YCbCr420 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **aSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToRGB_JPEG_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int
aSrcStep[3], *Npp8u* *pDst, int nDstStep,
NppiSize oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **aSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToRGB_JPEG_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int
aSrcStep[3], *Npp8u* *pDst, int nDstStep,
NppiSize oSizeROI, *NppStreamContext*
nppStreamCtx)

3 channel 8-bit unsigned packed YCbCr411 to 3 channel 8-bit unsigned planar RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr444ToRGB_JPEG_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr444 to 3 channel 8-bit unsigned packed RGB color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr420ToBGR_JPEG_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr422ToBGR_JPEG_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr411ToBGR_JPEG_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int aSrcStep[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **aSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiYCbCr444ToBGR_JPEG_8u_P3C3R_Ctx**(const *Npp8u* *const pSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar YCbCr444 to 3 channel 8-bit unsigned packed BGR color conversion.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.9. Image Data Exchange And Initialization Functions

Functions for initializing, copying and converting image data.

These functions can be found in the nppidei library. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

1.9.1. Set

Functions for setting all pixels within the ROI to a specific value.

1.9.1.1 Common parameters for nppiSet functions:

param nValue

The pixel value to be set.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiSet_8s_C1R_Ctx(const *Npp8s* nValue, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

8-bit image set.

For common parameter descriptions see *Common parameters for nppiSet functions*.

NppStatus nppiSet_8s_C2R_Ctx(const *Npp8s* aValue[2], *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

8-bit two-channel image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_8s_C3R_Ctx(const *Npp8s* aValue[3], *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

8-bit three-channel image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_8s_C4R_Ctx(const *Npp8s* aValue[4], *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

8-bit four-channel image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_8s_AC4R_Ctx(const *Npp8s* aValue[3], *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

8-bit four-channel image set ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_8u_C1R_Ctx(const *Npp8u* nValue, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

8-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_8u_C2R_Ctx(const *Npp8u* aValue[2], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 8-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_8u_C3R_Ctx(const *Npp8u* aValue[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_8u_C4R_Ctx(const *Npp8u* aValue[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_8u_AC4R_Ctx(const *Npp8u* aValue[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_16u_C1R_Ctx(const *Npp16u* nValue, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_16u_C2R_Ctx(const *Npp16u* aValue[2], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 16-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_16u_C3R_Ctx(const *Npp16u* aValue[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_16u_C4R_Ctx(const *Npp16u* aValue[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_16u_AC4R_Ctx(const *Npp16u* aValue[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_16s_C1R_Ctx(const *Npp16s* nValue, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_16s_C2R_Ctx(const *Npp16s* aValue[2], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 16-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_16s_C3R_Ctx(const *Npp16s* aValue[3], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 16-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_16s_C4R_Ctx(const *Npp16s* aValue[4], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_16s_AC4R_Ctx(const *Npp16s* aValue[3], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_16sc_C1R_Ctx**(const *Npp16sc* oValue, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit complex integer image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_16sc_C2R_Ctx**(const *Npp16sc* aValue[2], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit complex integer two-channel image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_16sc_C3R_Ctx**(const *Npp16sc* aValue[3], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit complex integer three-channel image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_16sc_C4R_Ctx**(const *Npp16sc* aValue[4], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit complex integer four-channel image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_16sc_AC4R_Ctx**(const *Npp16sc* aValue[3], *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit complex integer four-channel image set ignoring alpha.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32s_C1R_Ctx**(const *Npp32s* nValue, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32s_C2R_Ctx**(const *Npp32s* aValue[2], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 32-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32s_C3R_Ctx**(const *Npp32s* aValue[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 32-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32s_C4R_Ctx**(const *Npp32s* aValue[4], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32s_AC4R_Ctx**(const *Npp32s* aValue[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_32u_C1R_Ctx(const *Npp32u* nValue, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_32u_C2R_Ctx(const *Npp32u* aValue[2], *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 32-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_32u_C3R_Ctx(const *Npp32u* aValue[3], *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 32-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_32u_C4R_Ctx(const *Npp32u* aValue[4], *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_32u_AC4R_Ctx(const *Npp32u* aValue[3], *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit unsigned image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_32sc_C1R_Ctx(const *Npp32sc* oValue, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit complex integer image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_32sc_C2R_Ctx(const *Npp32sc* aValue[2], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Two channel 32-bit complex integer image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_32sc_C3R_Ctx(const *Npp32sc* aValue[3], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit complex integer image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_32sc_C4R_Ctx(const *Npp32sc* aValue[4], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit complex integer image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus nppiSet_32sc_AC4R_Ctx(const *Npp32sc* aValue[3], *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit complex integer four-channel image set ignoring alpha.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_16f_C1R_Ctx**(const *Npp32f* nValue, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_16f_C2R_Ctx**(const *Npp32f* aValues[2], *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 16-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_16f_C3R_Ctx**(const *Npp32f* aValues[3], *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 16-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_16f_C4R_Ctx**(const *Npp32f* aValues[4], *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32f_C1R_Ctx**(const *Npp32f* nValue, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32f_C2R_Ctx**(const *Npp32f* aValue[2], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

2 channel 32-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32f_C2R**(const *Npp32f* aValue[2], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI)

2 channel 32-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32f_C3R_Ctx**(const *Npp32f* aValue[3], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32f_C4R_Ctx**(const *Npp32f* aValue[4], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32f_AC4R_Ctx**(const *Npp32f* aValue[3], *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32fc_C1R_Ctx**(const *Npp32fc* oValue, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit complex image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32fc_C2R_Ctx**(const *Npp32fc* aValue[2], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Two channel 32-bit complex image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32fc_C3R_Ctx**(const *Npp32fc* aValue[3], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit complex image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32fc_C4R_Ctx**(const *Npp32fc* aValue[4], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit complex image set.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

NppStatus **nppiSet_32fc_AC4R_Ctx**(const *Npp32fc* aValue[3], *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit complex four-channel image set ignoring alpha.

For common parameter descriptions, see *Common parameters for nppiSet functions*.

1.9.2. Masked Set

The masked set primitives have an additional “mask image” input. The mask controls which pixels within the ROI are set. For details see *Masked Operation*.

1.9.2.1 Common parameters for nppiSet_CXM functions:

param nValue

The pixel value to be set for single channel functions.

param aValue

The pixel-value to be set for multi-channel functions.

param pDst

Pointer *Destination-Image Pointer*.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pMask

Mask-Image Pointer.

param nMaskStep

Mask-Image Line Step.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiSet_8u_C1MR_Ctx**(*Npp8u* nValue, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 8-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

NppStatus **nppiSet_8u_C3MR_Ctx**(const *Npp8u* aValue[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 3 channel 8-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

NppStatus **nppiSet_8u_C4MR_Ctx**(const *Npp8u* aValue[4], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 4 channel 8-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

NppStatus **nppiSet_8u_AC4MR_Ctx**(const *Npp8u* aValue[3], *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 4 channel 8-bit unsigned image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

NppStatus **nppiSet_16u_C1MR_Ctx**(*Npp16u* nValue, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 16-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

NppStatus **nppiSet_16u_C3MR_Ctx**(const *Npp16u* aValue[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 3 channel 16-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

NppStatus nppiSet_16u_C4MR_Ctx(const *Npp16u* aValue[4], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 4 channel 16-bit unsigned image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*.

NppStatus nppiSet_16u_AC4MR_Ctx(const *Npp16u* aValue[3], *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 4 channel 16-bit unsigned image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*.

NppStatus nppiSet_16s_C1MR_Ctx(*Npp16s* nValue, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 16-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*.

NppStatus nppiSet_16s_C3MR_Ctx(const *Npp16s* aValue[3], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 3 channel 16-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*.

NppStatus nppiSet_16s_C4MR_Ctx(const *Npp16s* aValue[4], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 4 channel 16-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*.

NppStatus nppiSet_16s_AC4MR_Ctx(const *Npp16s* aValue[3], *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 4 channel 16-bit image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*.

NppStatus nppiSet_32s_C1MR_Ctx(*Npp32s* nValue, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 32-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*.

NppStatus nppiSet_32s_C3MR_Ctx(const *Npp32s* aValue[3], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 3 channel 32-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*.

NppStatus nppiSet_32s_C4MR_Ctx(const *Npp32s* aValue[4], *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked 4 channel 32-bit image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

```
NppStatus nppiSet_32s_AC4MR_Ctx( const Npp32s aValue[3], Npp32s *pDst, int nDstStep,  
                                   NppiSize oSizeROI, const Npp8u *pMask, int nMaskStep,  
                                   NppStreamContext nppStreamCtx)
```

Masked 4 channel 16-bit image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

```
NppStatus nppiSet_32f_C1MR_Ctx( Npp32f nValue, Npp32f *pDst, int nDstStep, NppiSize oSizeROI,  
                                   const Npp8u *pMask, int nMaskStep, NppStreamContext  
                                   nppStreamCtx)
```

Masked 32-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

```
NppStatus nppiSet_32f_C3MR_Ctx( const Npp32f aValue[3], Npp32f *pDst, int nDstStep, NppiSize  
                                   oSizeROI, const Npp8u *pMask, int nMaskStep,  
                                   NppStreamContext nppStreamCtx)
```

Masked 3 channel 32-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

```
NppStatus nppiSet_32f_C4MR_Ctx( const Npp32f aValue[4], Npp32f *pDst, int nDstStep, NppiSize  
                                   oSizeROI, const Npp8u *pMask, int nMaskStep,  
                                   NppStreamContext nppStreamCtx)
```

Masked 4 channel 32-bit floating point image set.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

```
NppStatus nppiSet_32f_AC4MR_Ctx( const Npp32f aValue[3], Npp32f *pDst, int nDstStep,  
                                   NppiSize oSizeROI, const Npp8u *pMask, int nMaskStep,  
                                   NppStreamContext nppStreamCtx)
```

Masked 4 channel 32-bit floating point image set method, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXM functions*:

1.9.3. Channel Set

The selected-channel set primitives set a single color channel in multi-channel images to a given value. The channel is selected by adjusting the pDst pointer to point to the desired color channel (see *Channel-of-Interest API*).

1.9.3.1 Common parameters for nppiSet_CXC functions:

param nValue

The pixel-value to be set.

param pDst

Select-Channel Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiSet_8u_C3CR_Ctx**(*Npp8u* nValue, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned image set affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXC functions:*

NppStatus **nppiSet_8u_C4CR_Ctx**(*Npp8u* nValue, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image set affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXC functions:*

NppStatus **nppiSet_16u_C3CR_Ctx**(*Npp16u* nValue, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned image set affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXC functions:*

NppStatus **nppiSet_16u_C4CR_Ctx**(*Npp16u* nValue, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned image set affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXC functions:*

NppStatus **nppiSet_16s_C3CR_Ctx**(*Npp16s* nValue, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed image set affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXC functions:*

NppStatus **nppiSet_16s_C4CR_Ctx**(*Npp16s* nValue, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed image set affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXC functions:*

NppStatus **nppiSet_32s_C3CR_Ctx**(*Npp32s* nValue, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 32-bit unsigned image set affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXC functions*.

NppStatus **nppiSet_32s_C4CR_Ctx**(*Npp32s* nValue, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit unsigned image set affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXC functions*.

NppStatus **nppiSet_32f_C3CR_Ctx**(*Npp32f* nValue, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image set affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXC functions*.

NppStatus **nppiSet_32f_C4CR_Ctx**(*Npp32f* nValue, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image set affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiSet_CXC functions*.

Functions for copying image pixels.

1.9.4. Copy

Copy pixels from one image to another.

1.9.4.1 Common parameters for nppiCopy functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiCopy_8s_C1R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

8-bit image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions:*.

NppStatus nppiCopy_8s_C2R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Two-channel 8-bit image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions:*.

NppStatus nppiCopy_8s_C3R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions:*.

NppStatus nppiCopy_8s_C4R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions:*.

NppStatus nppiCopy_8s_AC4R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit image copy, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiCopy functions:*.

NppStatus nppiCopy_8u_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

8-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions:*.

NppStatus nppiCopy_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions:*.

NppStatus nppiCopy_8u_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions:*.

NppStatus nppiCopy_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image copy, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiCopy functions:*.

NppStatus nppiCopy_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions:*.

NppStatus **nppiCopy_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus **nppiCopy_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus **nppiCopy_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned image copy, not affecting Alpha channel.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus **nppiCopy_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus **nppiCopy_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus **nppiCopy_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus **nppiCopy_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit image copy, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus **nppiCopy_16sc_C1R_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit complex image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus **nppiCopy_16sc_C2R_Ctx**(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Two-channel 16-bit complex image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_16sc_C3R_Ctx(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit complex image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_16sc_C4R_Ctx(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit complex image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_16sc_AC4R_Ctx(const *Npp16sc* *pSrc, int nSrcStep, *Npp16sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit complex image copy, ignoring alpha.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32s_C1R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32s_C3R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32s_C4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32s_AC4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit image copy, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32sc_C1R_Ctx(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit complex image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32sc_C2R_Ctx(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Two-channel 32-bit complex image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32sc_C3R_Ctx(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit complex image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32sc_C4R_Ctx(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit complex image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32sc_AC4R_Ctx(const *Npp32sc* *pSrc, int nSrcStep, *Npp32sc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit complex image copy, ignoring alpha.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_16f_C1R_Ctx(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

16-bit floating point image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_16f_C3R_Ctx(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit floating point image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_16f_C4R_Ctx(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit floating point image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating point image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image copy.

For common parameter descriptions, see *Common parameters for nppiCopy functions*.

NppStatus nppiCopy_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image copy, not affecting Alpha.

For common parameter descriptions, see [Common parameters for nppiCopy functions](#):

NppStatus **nppiCopy_32fc_C1R_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit floating-point complex image copy.

For common parameter descriptions, see [Common parameters for nppiCopy functions](#):

NppStatus **nppiCopy_32fc_C2R_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Two-channel 32-bit floating-point complex image copy.

For common parameter descriptions, see [Common parameters for nppiCopy functions](#):

NppStatus **nppiCopy_32fc_C3R_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating-point complex image copy.

For common parameter descriptions, see [Common parameters for nppiCopy functions](#):

NppStatus **nppiCopy_32fc_C4R_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point complex image copy.

For common parameter descriptions, see [Common parameters for nppiCopy functions](#):

NppStatus **nppiCopy_32fc_AC4R_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32fc* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point complex image copy, ignoring alpha.

For common parameter descriptions, see [Common parameters for nppiCopy functions](#):

1.9.5. Masked Copy

The masked copy primitives have an additional “mask image” input. The mask controls which pixels within the ROI are copied. For details see [Masked Operation](#).

1.9.5.1 Common parameters for nppiCopy_CXM functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param pMask
Mask-Image Pointer.

param nMaskStep
Mask-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiCopy_8u_C1MR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation 8-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions:*

NppStatus **nppiCopy_8u_C3MR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation three channel 8-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions:*

NppStatus **nppiCopy_8u_C4MR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation four channel 8-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions:*

NppStatus **nppiCopy_8u_AC4MR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation four channel 8-bit unsigned image copy, ignoring alpha.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions:*

NppStatus nppiCopy_16u_C1MR_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation 16-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*.

NppStatus nppiCopy_16u_C3MR_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation three channel 16-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*.

NppStatus nppiCopy_16u_C4MR_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation four channel 16-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*.

NppStatus nppiCopy_16u_AC4MR_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation four channel 16-bit unsigned image copy, ignoring alpha.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*.

NppStatus nppiCopy_16s_C1MR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation 16-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*.

NppStatus nppiCopy_16s_C3MR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation three channel 16-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*.

NppStatus nppiCopy_16s_C4MR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation four channel 16-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*.

NppStatus nppiCopy_16s_AC4MR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation four channel 16-bit signed image copy, ignoring alpha.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*.

NppStatus nppiCopy_32s_C1MR_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation 32-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*:

NppStatus **nppiCopy_32s_C3MR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation three channel 32-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*:

NppStatus **nppiCopy_32s_C4MR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation four channel 32-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*:

NppStatus **nppiCopy_32s_AC4MR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation four channel 32-bit signed image copy, ignoring alpha.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*:

NppStatus **nppiCopy_32f_C1MR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation 32-bit float image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*:

NppStatus **nppiCopy_32f_C3MR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation three channel 32-bit float image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*:

NppStatus **nppiCopy_32f_C4MR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation four channel 32-bit float image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*:

NppStatus **nppiCopy_32f_AC4MR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, int nMaskStep, *NppStreamContext* nppStreamCtx)

Masked Operation four channel 32-bit float image copy, ignoring alpha.

For common parameter descriptions, see *Common parameters for nppiCopy_CXM functions*:

1.9.6. Channel Copy

The channel copy primitives copy a single color channel from a multi-channel source image to any other color channel in a multi-channel destination image. The channel is selected by adjusting the respective image pointers to point to the desired color channel (see *Channel-of-Interest API*).

1.9.6.1 Common parameters for nppiCopy_CXC functions:

param pSrc
Select-Channel Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Select-Channel Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiCopy_8u_C3CR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Selected channel 8-bit unsigned image copy for three-channel images.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC functions*.

NppStatus **nppiCopy_8u_C4CR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Selected channel 8-bit unsigned image copy for four-channel images.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC functions*.

NppStatus **nppiCopy_16s_C3CR_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Selected channel 16-bit signed image copy for three-channel images.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC functions*.

NppStatus **nppiCopy_16s_C4CR_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Selected channel 16-bit signed image copy for four-channel images.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC functions*:

NppStatus **nppiCopy_16u_C3CR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Selected channel 16-bit unsigned image copy for three-channel images.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC functions*:

NppStatus **nppiCopy_16u_C4CR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Selected channel 16-bit unsigned image copy for four-channel images.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC functions*:

NppStatus **nppiCopy_32s_C3CR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Selected channel 32-bit signed image copy for three-channel images.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC functions*:

NppStatus **nppiCopy_32s_C4CR_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Selected channel 32-bit signed image copy for four-channel images.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC functions*:

NppStatus **nppiCopy_32f_C3CR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Selected channel 32-bit float image copy for three-channel images.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC functions*:

NppStatus **nppiCopy_32f_C4CR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Selected channel 32-bit float image copy for four-channel images.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC functions*:

1.9.7. Extract Channel Copy

The channel extract primitives copy a single color channel from a multi-channel source image to single-channel destination image. The channel is selected by adjusting the source image pointer to point to the desired color channel (see *Channel-of-Interest API*).

1.9.7.1 Common parameters for nppiCopy_CXC1 functions:

param pSrc

Select-Channel Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiCopy_8u_C3C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel to single-channel 8-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions:*

NppStatus nppiCopy_8u_C4C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel to single-channel 8-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions:*

NppStatus nppiCopy_16s_C3C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel to single-channel 16-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions:*

NppStatus nppiCopy_16s_C4C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel to single-channel 16-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions:*

NppStatus nppiCopy_16u_C3C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel to single-channel 16-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions:*

NppStatus **nppiCopy_16u_C4C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel to single-channel 16-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions*.

NppStatus **nppiCopy_32s_C3C1R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel to single-channel 32-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions*.

NppStatus **nppiCopy_32s_C4C1R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel to single-channel 32-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions*.

NppStatus **nppiCopy_32f_C2C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Two-channel to single-channel 32-bit float image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions*.

NppStatus **nppiCopy_32f_C3C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel to single-channel 32-bit float image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions*.

NppStatus **nppiCopy_32f_C4C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel to single-channel 32-bit float image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXC1 functions*.

1.9.8. Insert Channel Copy

The channel insert primitives copy a single-channel source image into one of the color channels in a multi-channel destination image. The channel is selected by adjusting the destination image pointer to point to the desired color channel (see *Channel-of-Interest API*).

1.9.8.1 Common parameters for nppiCopy_C1CX functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Select-Channel Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiCopy_8u_C1C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to three-channel 8-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions:*

NppStatus nppiCopy_8u_C1C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to four-channel 8-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions:*

NppStatus nppiCopy_16s_C1C3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to three-channel 16-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions:*

NppStatus nppiCopy_16s_C1C4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to four-channel 16-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions:*

NppStatus nppiCopy_16u_C1C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to three-channel 16-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions:*

NppStatus **nppiCopy_16u_C1C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to four-channel 16-bit unsigned image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions*.

NppStatus **nppiCopy_32s_C1C3R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to three-channel 32-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions*.

NppStatus **nppiCopy_32s_C1C4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to four-channel 32-bit signed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions*.

NppStatus **nppiCopy_32f_C1C2R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to two-channel 32-bit float image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions*.

NppStatus **nppiCopy_32f_C1C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to three-channel 32-bit float image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions*.

NppStatus **nppiCopy_32f_C1C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel to four-channel 32-bit float image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_C1CX functions*.

1.9.9. Packed To Planar Channel Copy

Split a packed multi-channel image into multiple single channel planes.

E.g. copy the three channels of an RGB image into three separate single-channel images.

1.9.9.1 Common parameters for nppiCopy_CXPX functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param aDst
Destination-Planar-Image Pointer Array.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiCopy_8u_C3P3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *const aDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned packed to planar image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXPX functions:*

NppStatus nppiCopy_8u_C4P4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *const aDst[4], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned packed to planar image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXPX functions:*

NppStatus nppiCopy_16s_C3P3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *const aDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed packed to planar image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXPX functions:*

NppStatus nppiCopy_16s_C4P4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *const aDst[4], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed packed to planar image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXPX functions:*

NppStatus nppiCopy_16u_C3P3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *const aDst[3], int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned packed to planar image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXPX functions*:

NppStatus **nppiCopy_16u_C4P4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *const aDst[4],
int nDstStep, *NppiSize* oSizeROI, *NppStreamContext*
nppStreamCtx)

Four-channel 16-bit unsigned packed to planar image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXPX functions*:

NppStatus **nppiCopy_32s_C3P3R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *const aDst[3],
int nDstStep, *NppiSize* oSizeROI, *NppStreamContext*
nppStreamCtx)

Three-channel 32-bit signed packed to planar image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXPX functions*:

NppStatus **nppiCopy_32s_C4P4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *const aDst[4],
int nDstStep, *NppiSize* oSizeROI, *NppStreamContext*
nppStreamCtx)

Four-channel 32-bit signed packed to planar image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXPX functions*:

NppStatus **nppiCopy_32f_C3P3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *const aDst[3],
int nDstStep, *NppiSize* oSizeROI, *NppStreamContext*
nppStreamCtx)

Three-channel 32-bit float packed to planar image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXPX functions*:

NppStatus **nppiCopy_32f_C4P4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *const aDst[4],
int nDstStep, *NppiSize* oSizeROI, *NppStreamContext*
nppStreamCtx)

Four-channel 32-bit float packed to planar image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_CXPX functions*:

1.9.10. Planar To Packed Channel Copy

Combine multiple image planes into a packed multi-channel image.

E.g. copy three single-channel images into a single 3-channel image.

1.9.10.1 Common parameters for nppiCopy_PXCX functions:

param aSrc
Planar *Source-Image Pointer*.

param nSrcStep
Source-Planar-Image Pointer Array.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiCopy_8u_P3C3R_Ctx(const *Npp8u* *const aSrc[3], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned planar to packed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_PXCX functions*.

NppStatus nppiCopy_8u_P4C4R_Ctx(const *Npp8u* *const aSrc[4], int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned planar to packed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_PXCX functions*.

NppStatus nppiCopy_16u_P3C3R_Ctx(const *Npp16u* *const aSrc[3], int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned planar to packed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_PXCX functions*.

NppStatus nppiCopy_16u_P4C4R_Ctx(const *Npp16u* *const aSrc[4], int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned planar to packed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_PXCX functions*.

NppStatus nppiCopy_16s_P3C3R_Ctx(const *Npp16s* *const aSrc[3], int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed planar to packed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_PXCX functions*:

NppStatus **nppiCopy_16s_P4C4R_Ctx**(const *Npp16s* *const aSrc[4], int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed planar to packed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_PXCX functions*:

NppStatus **nppiCopy_32s_P3C3R_Ctx**(const *Npp32s* *const aSrc[3], int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit signed planar to packed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_PXCX functions*:

NppStatus **nppiCopy_32s_P4C4R_Ctx**(const *Npp32s* *const aSrc[4], int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed planar to packed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_PXCX functions*:

NppStatus **nppiCopy_32f_P3C3R_Ctx**(const *Npp32f* *const aSrc[3], int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit float planar to packed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_PXCX functions*:

NppStatus **nppiCopy_32f_P4C4R_Ctx**(const *Npp32f* *const aSrc[4], int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit float planar to packed image copy.

For common parameter descriptions, see *Common parameters for nppiCopy_PXCX functions*:

1.9.11. Copy Constant Border

Methods for copying images and padding borders with a constant, user-specifiable color.

1.9.11.1 Common parameters for nppiCopyConstBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSizeROI

Size of the source region of pixels.

param pDst*Destination-Image Pointer.***param nDstStep***Destination-Image Line Step.***param oDstSizeROI**

Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

param nTopBorderHeightHeight (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.**param nLeftBorderWidth**Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.**param nValue**

The pixel value to be set for border pixels for single channel functions.

param aValue

Vector of the RGBA values of the border pixels to be set for multi-channel functions.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes***Functions**

NppStatus **nppiCopyConstBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *Npp8u* nValue, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*.

NppStatus **nppiCopyConstBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp8u* aValue[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*.

NppStatus **nppiCopyConstBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp8u* aValue[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions:*.

NppStatus **nppiCopyConstBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp8u* aValue[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer image copy with constant border color with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions:*.

NppStatus **nppiCopyConstBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *Npp16u* nValue, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions:*.

NppStatus **nppiCopyConstBorder_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp16u* aValue[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions:*.

NppStatus **nppiCopyConstBorder_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp16u* aValue[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions:*.

NppStatus **nppiCopyConstBorder_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp16u* aValue[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned integer image copy with constant border color with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions:*.

NppStatus nppiCopyConstBorder_16s_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *Npp16s* nValue, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*.

NppStatus nppiCopyConstBorder_16s_C3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp16s* aValue[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*.

NppStatus nppiCopyConstBorder_16s_C4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp16s* aValue[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*.

NppStatus nppiCopyConstBorder_16s_AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp16s* aValue[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer image copy with constant border color with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*.

NppStatus nppiCopyConstBorder_32s_C1R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *Npp32s* nValue, *NppStreamContext* nppStreamCtx)

1 channel 32-bit signed integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*.

NppStatus nppiCopyConstBorder_32s_C3R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp32s* aValue[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit signed integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*:

NppStatus nppiCopyConstBorder_32s_C4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp32s* aValue[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*:

NppStatus nppiCopyConstBorder_32s_AC4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp32s* aValue[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer image copy with constant border color with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*:

NppStatus nppiCopyConstBorder_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *Npp32f* nValue, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*:

NppStatus nppiCopyConstBorder_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp32f* aValue[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*:

NppStatus nppiCopyConstBorder_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp32f* aValue[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image copy with constant border color.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*:

NppStatus nppiCopyConstBorder_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const *Npp32f* aValue[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image copy with constant border color with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyConstBorder functions*.

1.9.12. Copy Replicate Border

Methods for copying images and padding borders with a replicates of the nearest source image pixel color.

1.9.12.1 Common parameters for nppiCopyReplicateBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSizeROI

Size of the source region of pixels.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstSizeROI

Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).

param nTopBorderHeight

Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

param nLeftBorderWidth

Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiCopyReplicateBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*.

NppStatus **nppiCopyReplicateBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*.

NppStatus **nppiCopyReplicateBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*.

NppStatus **nppiCopyReplicateBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer image copy with nearest source image pixel color with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*.

NppStatus **nppiCopyReplicateBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*.

NppStatus **nppiCopyReplicateBorder_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions:*

NppStatus **nppiCopyReplicateBorder_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions:*

NppStatus **nppiCopyReplicateBorder_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned image copy with nearest source image pixel color with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions:*

NppStatus **nppiCopyReplicateBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions:*

NppStatus **nppiCopyReplicateBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions:*

NppStatus **nppiCopyReplicateBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions:*

NppStatus nppiCopyReplicateBorder_16s_AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer image copy with nearest source image pixel color with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*.

NppStatus nppiCopyReplicateBorder_32s_C1R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 32-bit signed integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*.

NppStatus nppiCopyReplicateBorder_32s_C3R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

3 channel 32-bit signed image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*.

NppStatus nppiCopyReplicateBorder_32s_C4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*.

NppStatus nppiCopyReplicateBorder_32s_AC4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer image copy with nearest source image pixel color with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*.

NppStatus nppiCopyReplicateBorder_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*:

NppStatus **nppiCopyReplicateBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*:

NppStatus **nppiCopyReplicateBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image copy with nearest source image pixel color.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*:

NppStatus **nppiCopyReplicateBorder_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image copy with nearest source image pixel color with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyReplicateBorder functions*:

1.9.13. Copy Wrap Border

Methods for copying images and padding borders with wrapped replications of the source image pixel colors.

1.9.13.1 Common parameters for nppiCopyWrapBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSizeROI

Size of the source region of pixels.

param pDst*Destination-Image Pointer.***param nDstStep***Destination-Image Line Step.***param oDstSizeROI**

Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

param nTopBorderHeight

Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors. $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$.

param nLeftBorderWidth

Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI: $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes*

Functions

NppStatus **nppiCopyWrapBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*.

NppStatus **nppiCopyWrapBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*.

NppStatus **nppiCopyWrapBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*:

NppStatus nppiCopyWrapBorder_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*:

NppStatus nppiCopyWrapBorder_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*:

NppStatus nppiCopyWrapBorder_16u_C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*:

NppStatus nppiCopyWrapBorder_16u_C4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*:

NppStatus nppiCopyWrapBorder_16u_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions:*.

NppStatus nppiCopyWrapBorder_16s_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions:*.

NppStatus nppiCopyWrapBorder_16s_C3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions:*.

NppStatus nppiCopyWrapBorder_16s_C4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions:*.

NppStatus nppiCopyWrapBorder_16s_AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions:*.

NppStatus nppiCopyWrapBorder_32s_C1R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions:*.

NppStatus nppiCopyWrapBorder_32s_C3R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

3 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*.

NppStatus nppiCopyWrapBorder_32s_C4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*.

NppStatus nppiCopyWrapBorder_32s_AC4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*.

NppStatus nppiCopyWrapBorder_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*.

NppStatus nppiCopyWrapBorder_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*.

NppStatus **nppiCopyWrapBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*.

NppStatus **nppiCopyWrapBorder_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSizeROI, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopyWrapBorder functions*.

1.9.14. Copy Sub-Pixel

Functions for copying linearly interpolated images using source image subpixel coordinates.

1.9.14.1 Common parameters for nppiCopySubPix functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstSizeROI

Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

param nDx

Fractional part of source image X coordinate.

param nDy

Fractional part of source image Y coordinate.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiCopySubpix_8u_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*.

NppStatus nppiCopySubpix_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*.

NppStatus nppiCopySubpix_8u_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*.

NppStatus nppiCopySubpix_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*.

NppStatus nppiCopySubpix_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*.

NppStatus nppiCopySubpix_16u_C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*.

NppStatus nppiCopySubpix_16u_C4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*:

NppStatus **nppiCopySubpix_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*:

NppStatus **nppiCopySubpix_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*:

NppStatus **nppiCopySubpix_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*:

NppStatus **nppiCopySubpix_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*:

NppStatus **nppiCopySubpix_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*:

NppStatus **nppiCopySubpix_32s_C1R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

1 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*:

NppStatus **nppiCopySubpix_32s_C3R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

3 channel 32-bit signed linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions*:

NppStatus **nppiCopySubpix_32s_C4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions:*.

NppStatus **nppiCopySubpix_32s_AC4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions:*.

NppStatus **nppiCopySubpix_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions:*.

NppStatus **nppiCopySubpix_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions:*.

NppStatus **nppiCopySubpix_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions:*.

NppStatus **nppiCopySubpix_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *Npp32f* nDx, *Npp32f* nDy, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiCopySubPix functions:*.

1.9.15. Convert Bit Depth

Functions for converting bit depth without scaling.

1.9.16. Convert To Increased Bit Depth

The integer conversion methods do not involve any scaling. Also, even when increasing the bit-depth loss of information may occur:

- ▶ When converting integers (e.g. `Npp32u`) to float (e.g. `Npp32f`) integer value not accurately representable by the float are rounded to the closest floating-point value.
- ▶ When converting signed integers to unsigned integers all negative values are lost (saturated to 0).

Note that all pointers and step sizes for images with 16f (*Npp16f*) data types perform best when they are at least 16 byte aligned.

1.9.16.1 Common parameters for `nppiConvert` to increased bit depth functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiConvert_8u16u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions:*.

NppStatus **nppiConvert_8u16u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions:*.

NppStatus **nppiConvert_8u16u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u16u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u16s_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u16s_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u16s_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u32s_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u32s_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u32s_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u32s_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u32f_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u32f_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8u32f_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8s32s_C1R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_8s32s_C3R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit signed to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_8s32s_C4R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit signed to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_8s32s_AC4R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit signed to 32-bit signed conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_8s32f_C1R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_8s32f_C3R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit signed to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_8s32f_C4R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit signed to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_8s32f_AC4R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit signed to 32-bit floating-point conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16u32s_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16u32s_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16u32s_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16u32s_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned to 32-bit signed conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16u32f_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16u32f_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16u32f_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16s32s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16s32s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16s32s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed to 32-bit signed conversion.

NppStatus **nppiConvert_16s32s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed to 32-bit signed conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16s32f_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16s32f_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16s32f_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16s32f_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed to 32-bit floating-point conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16f32f_C1R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit floating-point to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16f32f_C3R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit floating-point to 32-bit floating-point conversion.

Note that all pointers and step sizes for images with 16f (*Npp16f*) data types perform best when they are at least 16 byte aligned

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16f32f_C4R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit floating-point to 32-bit floating-point conversion.

Note that all pointers and step sizes for images with 16f (*Npp16f*) data types perform best when they are at least 16 byte aligned.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_16f32f_AC4R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit floating-point to 32-bit floating-point conversion, not affecting Alpha.

Note that all pointers and step sizes for images with 16f (*Npp16f*) data types perform best when they are at least 16 byte aligned.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus **nppiConvert_8s8u_C1Rs_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 8-bit unsigned conversion with saturation.

Note that all pointers and step sizes for images with 16f (*Npp16f*) data types perform best when they are at least 16 byte aligned.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_8s16u_C1Rs_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit unsigned conversion with saturation.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_8s16s_C1R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_8s32u_C1Rs_Ctx(const *Npp8s* *pSrc, int nSrcStep, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 32-bit unsigned conversion with saturation.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_16s16u_C1Rs_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed to 16-bit unsigned conversion with saturation.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_16s32u_C1Rs_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed to 32-bit unsigned conversion with saturation.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_16u32u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned to 32-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_32s32u_C1Rs_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit signed to 32-bit unsigned conversion with saturation.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*.

NppStatus nppiConvert_32s32f_C1R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit signed to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*:

NppStatus **nppiConvert_32u32f_C1R_Ctx**(const *Npp32u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit unsigned to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to increased bit depth functions*:

1.9.17. Convert To Decreased Bit Depth

The integer conversion methods do not involve any scaling. When converting floating-point values to integers the user may choose the most appropriate rounding-mode. Typically information is lost when converting to lower bit depth:

- ▶ All converted values are saturated to the destination type's range. E.g. any values larger than the largest value of the destination type are clamped to the destination's maximum.
- ▶ Converting floating-point values to integer also involves rounding, effectively losing all fractional value information in the process.

Note that all pointers and step sizes for images with 16f (*Npp16f*) data types perform best when they are at least 16 byte aligned.

1.9.17.1 Common parameters for nppiConvert to decreased bit depth functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eRoundMode

Rounding Mode Parameter.

param nScaleFactor

Integer Result Scaling.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiConvert_16u8u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_16u8u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_16u8u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_16u8u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_16s8u_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_16s8u_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_16s8u_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_16s8u_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32s8u_C1R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32s8u_C3R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32s8u_C4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32s8u_AC4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32s8s_C1R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit signed to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32s8s_C3R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit signed to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_32s8s_C4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit signed to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_32s8s_AC4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit signed to 8-bit signed conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_8u8s_C1RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_16u8s_C1RSfs_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_16s8s_C1RSfs_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_16u16s_C1RSfs_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_32u8u_C1RSfs_Ctx(const *Npp32u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Single channel 32-bit unsigned to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32u8s_C1RSfs_Ctx( const Npp32u *pSrc, int nSrcStep, Npp8s *pDst, int  
                                         nDstStep, NppiSize oSizeROI, NppRoundMode  
                                         eRoundMode, int nScaleFactor, NppStreamContext  
                                         nppStreamCtx )
```

Single channel 32-bit unsigned to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32u16u_C1RSfs_Ctx( const Npp32u *pSrc, int nSrcStep, Npp16u *pDst,  
                                           int nDstStep, NppiSize oSizeROI, NppRoundMode  
                                           eRoundMode, int nScaleFactor, NppStreamContext  
                                           nppStreamCtx )
```

Single channel 32-bit unsigned to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32u16s_C1RSfs_Ctx( const Npp32u *pSrc, int nSrcStep, Npp16s *pDst,  
                                           int nDstStep, NppiSize oSizeROI, NppRoundMode  
                                           eRoundMode, int nScaleFactor, NppStreamContext  
                                           nppStreamCtx )
```

Single channel 32-bit unsigned to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32u32s_C1RSfs_Ctx( const Npp32u *pSrc, int nSrcStep, Npp32s *pDst,  
                                           int nDstStep, NppiSize oSizeROI, NppRoundMode  
                                           eRoundMode, int nScaleFactor, NppStreamContext  
                                           nppStreamCtx )
```

Single channel 32-bit unsigned to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32s16u_C1RSfs_Ctx( const Npp32s *pSrc, int nSrcStep, Npp16u *pDst,  
                                           int nDstStep, NppiSize oSizeROI, NppRoundMode  
                                           eRoundMode, int nScaleFactor, NppStreamContext  
                                           nppStreamCtx )
```

Single channel 32-bit signed to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32s16s_C1RSfs_Ctx( const Npp32s *pSrc, int nSrcStep, Npp16s *pDst,  
                                           int nDstStep, NppiSize oSizeROI, NppRoundMode  
                                           eRoundMode, int nScaleFactor, NppStreamContext  
                                           nppStreamCtx )
```

Single channel 32-bit signed to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus **nppiConvert_32f8u_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating point to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f8u_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating point to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f8u_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f8u_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f8s_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating point to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f8s_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating point to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f8s_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f8s_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 8-bit signed conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f16u_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating point to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f16u_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating point to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f16u_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f16u_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 16-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f16s_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating point to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f16s_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating point to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions*.

NppStatus **nppiConvert_32f16s_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_32f16s_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 16-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_32f16f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating point to 16-bit floating-point conversion.

Note that all pointers and step sizes for images with 16f (*Npp16f*) data types perform best when they are at least 16 byte aligned.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_32f16f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating point to 16-bit floating-point conversion.

Note that all pointers and step sizes for images with 16f (*Npp16f*) data types perform best when they are at least 16 byte aligned.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_32f16f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 16-bit floating-point conversion.

Note that all pointers and step sizes for images with 16f (*Npp16f*) data types perform best when they are at least 16 byte aligned.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_32f16f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 16-bit floating-point conversion.

Note that all pointers and step sizes for images with 16f (*Npp16f*) data types perform best when they are at least 16 byte aligned.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

NppStatus nppiConvert_32f8u_C1RSfs_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating point to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32f8s_C1RSfs_Ctx(const Npp32f *pSrc, int nSrcStep, Npp8s *pDst, int  
nDstStep, NppiSize oSizeROI, NppRoundMode  
eRoundMode, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Single channel 32-bit floating point to 8-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32f16u_C1RSfs_Ctx(const Npp32f *pSrc, int nSrcStep, Npp16u *pDst,  
int nDstStep, NppiSize oSizeROI, NppRoundMode  
eRoundMode, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Single channel 32-bit floating point to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32f16s_C1RSfs_Ctx(const Npp32f *pSrc, int nSrcStep, Npp16s *pDst,  
int nDstStep, NppiSize oSizeROI, NppRoundMode  
eRoundMode, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Single channel 32-bit floating point to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32f32u_C1RSfs_Ctx(const Npp32f *pSrc, int nSrcStep, Npp32u *pDst,  
int nDstStep, NppiSize oSizeROI, NppRoundMode  
eRoundMode, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Single channel 32-bit floating point to 32-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

```
NppStatus nppiConvert_32f32s_C1RSfs_Ctx(const Npp32f *pSrc, int nSrcStep, Npp32s *pDst,  
int nDstStep, NppiSize oSizeROI, NppRoundMode  
eRoundMode, int nScaleFactor, NppStreamContext  
nppStreamCtx)
```

Single channel 32-bit floating point to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiConvert to decreased bit depth functions:*.

1.9.18. Scale Bit Depth

Functions for scaling bit depth up or down.

1.9.19. Scale To Higher Bit Depth

Functions for scaling images to higher bit depth.

To map source pixel `srcPixelValue` to destination pixel `dstPixelValue` the following equation is used:

$$\text{dstPixelValue} = \text{dstMinRangeValue} + \text{scaleFactor} * (\text{srcPixelValue} - \text{srcMinRangeValue})$$

where $\text{scaleFactor} = (\text{dstMaxRangeValue} - \text{dstMinRangeValue}) / (\text{srcMaxRangeValue} - \text{srcMinRangeValue})$.

For conversions between integer data types, the entire integer numeric range of the input data type is mapped onto the entire integer numeric range of the output data type.

For conversions to floating point data types the floating point data range is defined by the user supplied floating point values of `nMax` and `nMin` which are used as the `dstMaxRangeValue` and `dstMinRangeValue` respectively in the `scaleFactor` and `dstPixelValue` calculations and also as the saturation values to which output data is clamped.

When converting from floating-point values to integer values, `nMax` and `nMin` are used as the `srcMaxRangeValue` and `srcMinRangeValue` respectively in the `scaleFactor` and `dstPixelValue` calculations. Output values are saturated and clamped to the full output integer pixel value range.

1.9.19.1 Common parameters for `npPiScale` to higher bit depth functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nMin

specifies the minimum saturation value to which every output value will be clamped.

param nMax

specifies the maximum saturation value to which every output value will be clamped.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if `nMax <= nMin`.

Functions

NppStatus **nppiScale_8u16u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u16u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u16u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 16-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u16u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u16s_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u16s_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 16-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u16s_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u32s_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u32s_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u32s_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 32-bit signed conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u32s_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nMin, *Npp32f* nMax, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*.

NppStatus **nppiScale_8u32f_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nMin, *Npp32f* nMax, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*:

NppStatus **nppiScale_8u32f_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nMin, *Npp32f* nMax, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 32-bit floating-point conversion.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*:

NppStatus **nppiScale_8u32f_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nMin, *Npp32f* nMax, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiScale to higher bit depth functions*:

1.9.20. Scale To Lower Bit Depth

Functions for scaling images to lower bit depth.

To map source pixel srcPixelValue to destination pixel dstPixelValue the following equation is used:

$$\text{dstPixelValue} = \text{dstMinRangeValue} + \text{scaleFactor} * (\text{srcPixelValue} - \text{srcMinRangeValue})$$

where $\text{scaleFactor} = (\text{dstMaxRangeValue} - \text{dstMinRangeValue}) / (\text{srcMaxRangeValue} - \text{srcMinRangeValue})$.

For conversions between integer data types, the entire integer numeric range of the input data type is mapped onto the entire integer numeric range of the output data type.

For conversions to floating point data types the floating point data range is defined by the user supplied floating point values of nMax and nMin which are used as the dstMaxRangeValue and dstMinRangeValue respectively in the scaleFactor and dstPixelValue calculations and also as the saturation values to which output data is clamped.

When converting from floating-point values to integer values, nMax and nMin are used as the srcMaxRangeValue and srcMinRangeValue respectively in the scaleFactor and dstPixelValue calculations. Output values are saturated and clamped to the full output integer pixel value range.

1.9.20.1 Common parameters for nppiScale to lower bit depth functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI*Region-Of-Interest (ROI).***param hint**

algorithm performance or accuracy selector, currently ignored

param nMin

specifies the minimum saturation value to which every output value will be clamped.

param nMax

specifies the maximum saturation value to which every output value will be clamped.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes, NPP_SCALE_RANGE_ERROR*
indicates an error condition if nMax <= nMin.**Functions***NppStatus* **nppiScale_16u8u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions:**NppStatus* **nppiScale_16u8u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions:**NppStatus* **nppiScale_16u8u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions:**NppStatus* **nppiScale_16u8u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions:**NppStatus* **nppiScale_16s8u_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*:

NppStatus **nppiScale_16s8u_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*:

NppStatus **nppiScale_16s8u_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*:

NppStatus **nppiScale_16s8u_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*:

NppStatus **nppiScale_32s8u_C1R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Single channel 32-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*:

NppStatus **nppiScale_32s8u_C3R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Three channel 32-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*:

NppStatus **nppiScale_32s8u_C4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Four channel 32-bit signed to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*:

NppStatus **nppiScale_32s8u_AC4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppHintAlgorithm* hint, *NppStreamContext* nppStreamCtx)

Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*:

NppStatus **nppiScale_32f8u_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nMin, *Npp32f* nMax, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating point to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*.

NppStatus **nppiScale_32f8u_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nMin, *Npp32f* nMax, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating point to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*.

NppStatus **nppiScale_32f8u_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nMin, *Npp32f* nMax, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 8-bit unsigned conversion.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*.

NppStatus **nppiScale_32f8u_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nMin, *Npp32f* nMax, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.

For common parameter descriptions, see *Common parameters for nppiScale to lower bit depth functions*.

1.9.21. Duplicate Channel

Functions for duplicating a single channel image in a multiple channel image.

1.9.21.1 Common parameters for nppiDup functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstSizeROI

Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiDup_8u_C1C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer source image duplicated in all 3 channels of destination image.

For common parameter descriptions, see *Common parameters for nppiDup functions:*

NppStatus **nppiDup_8u_C1C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer source image duplicated in all 4 channels of destination image.

For common parameter descriptions, see *Common parameters for nppiDup functions:*

NppStatus **nppiDup_8u_C1AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiDup functions:*

NppStatus **nppiDup_16u_C1C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer source image duplicated in all 3 channels of destination image.

For common parameter descriptions, see *Common parameters for nppiDup functions:*

NppStatus **nppiDup_16u_C1C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer source image duplicated in all 4 channels of destination image.

For common parameter descriptions, see *Common parameters for nppiDup functions:*

NppStatus **nppiDup_16u_C1AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiDup functions:*

NppStatus **nppiDup_16s_C1C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed integer source image duplicated in all 3 channels of destination image.

For common parameter descriptions, see *Common parameters for nppiDup functions:*

NppStatus nppiDup_16s_C1C4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed integer source image duplicated in all 4 channels of destination image.

For common parameter descriptions, see *Common parameters for nppiDup functions*.

NppStatus nppiDup_16s_C1AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiDup functions*.

NppStatus nppiDup_32s_C1C3R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 32-bit signed integer source image duplicated in all 3 channels of destination image.

For common parameter descriptions, see *Common parameters for nppiDup functions*.

NppStatus nppiDup_32s_C1C4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 32-bit signed integer source image duplicated in all 4 channels of destination image.

For common parameter descriptions, see *Common parameters for nppiDup functions*.

NppStatus nppiDup_32s_C1AC4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 32-bit signed integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiDup functions*.

NppStatus nppiDup_32f_C1C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point source image duplicated in all 3 channels of destination image.

For common parameter descriptions, see *Common parameters for nppiDup functions*.

NppStatus nppiDup_32f_C1C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point source image duplicated in all 4 channels of destination image.

For common parameter descriptions, see *Common parameters for nppiDup functions*.

NppStatus nppiDup_32f_C1AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

For common parameter descriptions, see *Common parameters for nppiDup functions*.

1.9.22. Transpose

Functions for transposing images of various types. Like matrix transpose, image transpose is a mirror along the image's diagonal (upper-left to lower-right corner).

1.9.22.1 Common parameters for nppiTranspose functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Pointer to the destination ROI.

param nDstStep
Destination-Image Line Step.

param oSrcROI
Region-Of-Interest (ROI).

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiTranspose_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions:*.

NppStatus **nppiTranspose_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions:*.

NppStatus **nppiTranspose_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions:*.

NppStatus **nppiTranspose_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*:

NppStatus **nppiTranspose_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*:

NppStatus **nppiTranspose_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*:

NppStatus **nppiTranspose_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*:

NppStatus **nppiTranspose_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*:

NppStatus **nppiTranspose_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*:

NppStatus **nppiTranspose_32s_C1R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

1 channel 32-bit signed int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*:

NppStatus **nppiTranspose_32s_C3R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

3 channel 32-bit signed int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*:

NppStatus **nppiTranspose_32s_C4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed int image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*:

NppStatus **nppiTranspose_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*.

NppStatus **nppiTranspose_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*.

NppStatus **nppiTranspose_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSrcROI, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image transpose.

For common parameter descriptions, see *Common parameters for nppiTranspose functions*.

1.9.23. Swap Channels

Functions for swapping and duplicating channels in multiple channel images. The methods support arbitrary permutations of the original channels, including replication and setting one or more channels to a constant value.

Functions

NppStatus **nppiSwapChannels_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer source image to 3 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_8u_C3IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer in place image.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_8u_C4C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer source image to 3 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [2,1,0] converts this to a 3 channel BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer source image to 4 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.

- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, aDstOrder = [3,2,1,0] converts this to BGRA channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_8u_C4IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer in place image.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, aDstOrder = [3,2,1,0] converts this to BGRA channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_8u_C3C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], const *Npp8u* nValue, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer source image to 4 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [3,2,1,0] converts this to VBGR channel order.

- **nValue** – (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. nValue is either written or not written to a particular channel depending on the aDstOrder entry for that destination channel. An aDstOrder value of 3 will output nValue to that channel, an aDstOrder value greater than 3 will leave that particular destination channel value unmodified.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiSwapChannels_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer source image to 4 channel destination image with destination alpha channel unaffected.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [3,2,1,0] converts this to VBGR channel order. of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiSwapChannels_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned integer source image to 3 channel destination image.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16u_C3IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned integer in place image.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16u_C4C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned integer source image to 3 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [2,1,0] converts this to a 3 channel BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned integer source image to 4 channel destination image.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **pDst** – *Destination-Image Pointer*.
- **nDstStep** – *Destination-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, aDstOrder = [3,2,1,0] converts this to BGRA channel order.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16u_C4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned integer in place image.

Parameters

- **pSrcDst** – *In-Place Image Pointer*.
- **nSrcDstStep** – *In-Place-Image Line Step*.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, aDstOrder = [3,2,1,0] converts this to BGRA channel order.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16u_C3C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], const *Npp16u* nValue, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned integer source image to 4 channel destination image.

Parameters

- **pSrc** – *Source-Image Pointer*.

- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [3,2,1,0] converts this to VBGR channel order.
- **nValue** – (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. nValue is either written or not written to a particular channel depending on the aDstOrder entry for that destination channel. An aDstOrder value of 3 will output nValue to that channel, an aDstOrder value greater than 3 will leave that particular destination channel value unmodified.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiSwapChannels_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned integer source image to 4 channel destination image with destination alpha channel unaffected.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiSwapChannels_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed integer source image to 3 channel destination image.

Parameters

- **pSrc** – *Source-Image Pointer.*

- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16s_C3IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed integer in place image.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *In-Place-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16s_C4C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer source image to 3 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [2,1,0] converts this to a 3 channel BGR channel order.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer source image to 4 channel destination image.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, aDstOrder = [3,2,1,0] converts this to BGRA channel order.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16s_C4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer in place image.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, aDstOrder = [3,2,1,0] converts this to BGRA channel order.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16s_C3C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], const *Npp16s* nValue, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed integer source image to 4 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [3,2,1,0] converts this to VBGR channel order.
- ▶ **nValue** – (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. nValue is either written or not written to a particular channel depending on the aDstOrder entry for that destination channel. An aDstOrder value of 3 will output nValue to that channel, an aDstOrder value greater than 3 will leave that particular destination channel value unmodified.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed integer source image to 4 channel destination image with destination alpha channel unaffected.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_32s_C3R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit signed integer source image to 3 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_32s_C3IR_Ctx**(*Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit signed integer in place image.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_32s_C4C3R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer source image to 3 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [2,1,0] converts this to a 3 channel BGR channel order.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiSwapChannels_32s_C4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer source image to 4 channel destination image.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, aDstOrder = [3,2,1,0] converts this to BGRA channel order.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiSwapChannels_32s_C4IR_Ctx**(*Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed integer in place image.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, aDstOrder = [3,2,1,0] converts this to BGRA channel order.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

```
NppStatus nppiSwapChannels_32s_C3C4R_Ctx(const Npp32s *pSrc, int nSrcStep, Npp32s *pDst,  
int nDstStep, NppiSize oSizeROI, const int  
aDstOrder[4], const Npp32s nValue,  
NppStreamContext nppStreamCtx)
```

3 channel 32-bit signed integer source image to 4 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [3,2,1,0] converts this to VBGR channel order.
- ▶ **nValue** – (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. nValue is either written or not written to a particular channel depending on the aDstOrder entry for that destination channel. An aDstOrder value of 3 will output nValue to that channel, an aDstOrder value greater than 3 will leave that particular destination channel value unmodified.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiSwapChannels_32s_AC4R_Ctx(const Npp32s *pSrc, int nSrcStep, Npp32s *pDst,  
int nDstStep, NppiSize oSizeROI, const int  
aDstOrder[3], NppStreamContext nppStreamCtx)
```

4 channel 32-bit signed integer source image to 4 channel destination image with destination alpha channel unaffected.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiSwapChannels_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point source image to 3 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiSwapChannels_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point in place image.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – oSizeROI *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [2,1,0] converts this to BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiSwapChannels_32f_C4C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point source image to 3 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [2,1,0] converts this to a 3 channel BGR channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point source image to 4 channel destination image.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, aDstOrder = [3,2,1,0] converts this to BGRA channel order.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_32f_C4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place image.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that

is stored in the n-th channel of the output image. E.g. Given an ARGB image, aDstOrder = [3,2,1,0] converts this to BGRA channel order.

► **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_32f_C3C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[4], const *Npp32f* nValue, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point source image to 4 channel destination image.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, aDstOrder = [3,2,1,0] converts this to VBGR channel order.
- **nValue** – (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. nValue is either written or not written to a particular channel depending on the aDstOrder entry for that destination channel. An aDstOrder value of 3 will output nValue to that channel, an aDstOrder value greater than 3 will leave that particular destination channel value unmodified.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSwapChannels_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const int aDstOrder[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point source image to 4 channel destination image with destination alpha channel unaffected.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*

- ▶ **aDstOrder** – Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, aDstOrder = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10. Image Filtering Functions

Linear and non-linear image filtering functions.

Filtering functions are classified as *Neighborhood Operations*. It is the user's responsibility to avoid *Sampling Beyond Image Boundaries*.

These functions can be found in the nppif library. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

1.10.1. Image 1D Linear Filters

1.10.1.1 1DLinearFilter

The set of 1D linear filtering functions available in the library.

1.10.1.2 Image Filter Column

1.10.1.2.1 FilterColumn

Apply convolution filter with user specified 1D column of weights.

1.10.1.2.1.1 Common parameters for nppiFilterColumn functions:

Result pixel is equal to the sum of the products between the kernel coefficients (pKernel array) and corresponding neighboring column pixel values in the source image defined by nKernelDim and nAnchorY, divided by nDivisor.

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst*Destination-Image Pointer.***param nDstStep***Destination-Image Line Step.***param oROI***Region-Of-Interest (ROI).***param pKernel**

Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param nMaskSize

Length of the linear kernel array.

param nAnchor

Y offset of the kernel origin frame of reference relative to the source pixel.

param nDivisor

The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes***Functions**

NppStatus **nppiFilterColumn_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

8-bit unsigned single-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions:*

NppStatus **nppiFilterColumn_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

8-bit unsigned three-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions:*

NppStatus **nppiFilterColumn_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

8-bit unsigned four-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions:*

NppStatus **nppiFilterColumn_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

8-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*.

NppStatus **nppiFilterColumn_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit unsigned single-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*.

NppStatus **nppiFilterColumn_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit unsigned three-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*.

NppStatus **nppiFilterColumn_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit unsigned four-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*.

NppStatus **nppiFilterColumn_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*.

NppStatus **nppiFilterColumn_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit single-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*.

NppStatus **nppiFilterColumn_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit three-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*:

NppStatus **nppiFilterColumn_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit four-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*:

NppStatus **nppiFilterColumn_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit four-channel 1D column convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*:

NppStatus **nppiFilterColumn_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

32-bit float single-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*:

NppStatus **nppiFilterColumn_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

32-bit float three-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*:

NppStatus **nppiFilterColumn_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

32-bit float four-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*:

NppStatus **nppiFilterColumn_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

32-bit float four-channel 1D column convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*:

NppStatus **nppiFilterColumn_64f_C1R_Ctx**(const *Npp64f* *pSrc, *Npp32s* nSrcStep, *Npp64f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp64f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

64-bit float single-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn functions*:

1.10.1.3 Image Filter Column Border

1.10.1.3.1 FilterColumnBorder

General purpose 1D convolution column filter with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.1.3.1.1 Common parameters for nppiFilterColumnBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pKernel

Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param nMaskSize

Width of the kernel.

param nAnchor

X offset of the kernel origin frame of reference relative to the source pixel.

param nDivisor

The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterColumnBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions:*

NppStatus **nppiFilterColumnBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions:*

NppStatus **nppiFilterColumnBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel channel 8-bit unsigned 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions:*

NppStatus **nppiFilterColumnBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned convolution 1D column filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

```
NppStatus nppiFilterColumnBorder_16u_C1R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp16u *pDst, Npp32s nDstStep, NppiSize  
                                              oSizeROI, const Npp32s *pKernel, Npp32s  
                                              nMaskSize, Npp32s nAnchor, Npp32s  
                                              nDivisor, NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Single channel 16-bit unsigned convolution 1D column filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

```
NppStatus nppiFilterColumnBorder_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp16u *pDst, Npp32s nDstStep, NppiSize  
                                              oSizeROI, const Npp32s *pKernel, Npp32s  
                                              nMaskSize, Npp32s nAnchor, Npp32s  
                                              nDivisor, NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Three channel 16-bit unsigned 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

```
NppStatus nppiFilterColumnBorder_16u_C4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp16u *pDst, Npp32s nDstStep, NppiSize  
                                              oSizeROI, const Npp32s *pKernel, Npp32s  
                                              nMaskSize, Npp32s nAnchor, Npp32s  
                                              nDivisor, NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Four channel channel 16-bit 1D column unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

```
NppStatus nppiFilterColumnBorder_16u_AC4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp16u *pDst, Npp32s nDstStep, NppiSize  
                                              oSizeROI, const Npp32s *pKernel, Npp32s  
                                              nMaskSize, Npp32s nAnchor, Npp32s  
                                              nDivisor, NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned 1D column convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

NppStatus nppiFilterColumnBorder_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

NppStatus nppiFilterColumnBorder_16s_C3R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

NppStatus nppiFilterColumnBorder_16s_C4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

NppStatus nppiFilterColumnBorder_16s_AC4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit 1D column convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

NppStatus nppiFilterColumnBorder_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit float 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

```
NppStatus nppiFilterColumnBorder_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp32f *pDst, Npp32s nDstStep, NppiSize  
oSizeROI, const Npp32f *pKernel, Npp32s  
nMaskSize, Npp32s nAnchor,  
                                              NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Three channel 32-bit float 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

```
NppStatus nppiFilterColumnBorder_32f_C4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp32f *pDst, Npp32s nDstStep, NppiSize  
oSizeROI, const Npp32f *pKernel, Npp32s  
nMaskSize, Npp32s nAnchor,  
                                              NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Four channel 32-bit float 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

```
NppStatus nppiFilterColumnBorder_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp32f *pDst, Npp32s nDstStep, NppiSize  
oSizeROI, const Npp32f *pKernel, Npp32s  
nMaskSize, Npp32s nAnchor,  
                                              NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Four channel 32-bit float 1D column convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder functions*.

1.10.1.4 Image Filter Column 32f

1.10.1.4.1 FilterColumn32f

FilterColumn using floating-point weights.

1.10.1.4.1.1 Common parameters for nppiFilterColumn32f functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oROI
Region-Of-Interest (ROI).

param pKernel
Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param nMaskSize
Length of the linear kernel array.

param nAnchor
Y offset of the kernel origin frame of reference relative to the source pixel.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterColumn32f_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

8-bit unsigned single-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*.

NppStatus **nppiFilterColumn32f_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

8-bit unsigned three-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*:

```
NppStatus nppiFilterColumn32f_8u_C4R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep, Npp8u
                                         *pDst, Npp32s nDstStep, NppiSize oROI, const
                                         Npp32f *pKernel, Npp32s nMaskSize, Npp32s
                                         nAnchor, NppStreamContext nppStreamCtx)
```

8-bit unsigned four-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*:

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFilterColumn32f_8u_AC4R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep, Npp8u
                                           *pDst, Npp32s nDstStep, NppiSize oROI, const
                                           Npp32f *pKernel, Npp32s nMaskSize, Npp32s
                                           nAnchor, NppStreamContext nppStreamCtx)
```

8-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*:

```
NppStatus nppiFilterColumn32f_16u_C1R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, Npp16u
                                           *pDst, Npp32s nDstStep, NppiSize oROI, const
                                           Npp32f *pKernel, Npp32s nMaskSize, Npp32s
                                           nAnchor, NppStreamContext nppStreamCtx)
```

16-bit unsigned single-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*:

```
NppStatus nppiFilterColumn32f_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, Npp16u
                                           *pDst, Npp32s nDstStep, NppiSize oROI, const
                                           Npp32f *pKernel, Npp32s nMaskSize, Npp32s
                                           nAnchor, NppStreamContext nppStreamCtx)
```

16-bit unsigned three-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*:

```
NppStatus nppiFilterColumn32f_16u_C4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, Npp16u
                                           *pDst, Npp32s nDstStep, NppiSize oROI, const
                                           Npp32f *pKernel, Npp32s nMaskSize, Npp32s
                                           nAnchor, NppStreamContext nppStreamCtx)
```

16-bit unsigned four-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*:

```
NppStatus nppiFilterColumn32f_16u_AC4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, Npp16u
                                           *pDst, Npp32s nDstStep, NppiSize oROI, const
                                           Npp32f *pKernel, Npp32s nMaskSize, Npp32s
                                           nAnchor, NppStreamContext nppStreamCtx)
```

16-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*.

```
NppStatus nppiFilterColumn32f_16s_C1R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp16s
                                             *pDst, Npp32s nDstStep, NppiSize oROI, const
                                             Npp32f *pKernel, Npp32s nMaskSize, Npp32s
                                             nAnchor, NppStreamContext nppStreamCtx)
```

16-bit single-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*.

```
NppStatus nppiFilterColumn32f_16s_C3R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp16s
                                             *pDst, Npp32s nDstStep, NppiSize oROI, const
                                             Npp32f *pKernel, Npp32s nMaskSize, Npp32s
                                             nAnchor, NppStreamContext nppStreamCtx)
```

16-bit three-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*.

```
NppStatus nppiFilterColumn32f_16s_C4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp16s
                                             *pDst, Npp32s nDstStep, NppiSize oROI, const
                                             Npp32f *pKernel, Npp32s nMaskSize, Npp32s
                                             nAnchor, NppStreamContext nppStreamCtx)
```

16-bit four-channel 1D column convolution.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*.

```
NppStatus nppiFilterColumn32f_16s_AC4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp16s
                                             *pDst, Npp32s nDstStep, NppiSize oROI, const
                                             Npp32f *pKernel, Npp32s nMaskSize, Npp32s
                                             nAnchor, NppStreamContext nppStreamCtx)
```

16-bit four-channel 1D column convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumn32f functions*.

1.10.1.5 Image Filter Column Border 32f

1.10.1.5.1 FilterColumnBorder32f

General purpose 1D column convolution filter using floating-point weights with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.1.5.1.1 Common parameters for nppiFilterColumnBorder32f functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pKernel

Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param nMaskSize

Width of the kernel.

param nAnchor

X offset of the kernel origin frame of reference relative to the source pixel.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

```
NppStatus nppiFilterColumnBorder32f_8u_C1R_Ctx( const Npp8u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp8u
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp32f *pKernel, Npp32s
nMaskSize, Npp32s nAnchor,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Single channel 8-bit unsigned 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*.

NppStatus nppiFilterColumnBorder32f_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*.

NppStatus nppiFilterColumnBorder32f_8u_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*.

NppStatus nppiFilterColumnBorder32f_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned 1D column convolution filter with border control, ignore alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*.

NppStatus nppiFilterColumnBorder32f_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*.

NppStatus nppiFilterColumnBorder32f_16u_C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*:

```
NppStatus nppiFilterColumnBorder32f_16u_C4R_Ctx(const Npp16u *pSrc, int nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp16u *pDst, int nDstStep, NppiSize  
                                                oSizeROI, const Npp32f *pKernel,  
                                                Npp32s nMaskSize, Npp32s nAnchor,  
                                                NppiBorderType eBorderType,  
                                                NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*:

```
NppStatus nppiFilterColumnBorder32f_16u_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint  
                                                oSrcOffset, Npp16u *pDst, int  
                                                nDstStep, NppiSize oSizeROI, const  
                                                Npp32f *pKernel, Npp32s nMaskSize,  
                                                Npp32s nAnchor, NppiBorderType  
                                                eBorderType, NppStreamContext  
                                                nppStreamCtx)
```

Four channel 16-bit unsigned 1D column convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*:

```
NppStatus nppiFilterColumnBorder32f_16s_C1R_Ctx(const Npp16s *pSrc, int nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp16s *pDst, int nDstStep, NppiSize  
                                                oSizeROI, const Npp32f *pKernel,  
                                                Npp32s nMaskSize, Npp32s nAnchor,  
                                                NppiBorderType eBorderType,  
                                                NppStreamContext nppStreamCtx)
```

Single channel 16-bit 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*:

```
NppStatus nppiFilterColumnBorder32f_16s_C3R_Ctx(const Npp16s *pSrc, int nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp16s *pDst, int nDstStep, NppiSize  
                                                oSizeROI, const Npp32f *pKernel,  
                                                Npp32s nMaskSize, Npp32s nAnchor,  
                                                NppiBorderType eBorderType,  
                                                NppStreamContext nppStreamCtx)
```

Three channel 16-bit 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*:

```
NppStatus nppiFilterColumnBorder32f_16s_C4R_Ctx(const Npp16s *pSrc, int nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp16s *pDst, int nDstStep, NppiSize
oSizeROI, const Npp32f *pKernel,
Npp32s nMaskSize, Npp32s nAnchor,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 16-bit 1D column convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*:

```
NppStatus nppiFilterColumnBorder32f_16s_AC4R_Ctx(const Npp16s *pSrc, int nSrcStep,
NppiSize oSrcSize, NppiPoint
oSrcOffset, Npp16s *pDst, int
nDstStep, NppiSize oSizeROI, const
Npp32f *pKernel, Npp32s nMaskSize,
Npp32s nAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Four channel 16-bit 1D column convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterColumnBorder32f functions*:

1.10.1.6 Image Filter Row

1.10.1.6.1 FilterRow

Apply convolution filter with user specified 1D row of weights.

Result pixel is equal to the sum of the products between the kernel coefficients (pKernel array) and corresponding neighboring row pixel values in the source image defined by nKernelDim and nAnchorX, divided by nDivisor.

1.10.1.6.1.1 Common parameters for nppiFilterRow functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oROI
Region-Of-Interest (ROI).

param pKernel

Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param nMaskSize

Length of the linear kernel array.

param nAnchor

X offset of the kernel origin frame of reference relative to the source pixel.

param nDivisor

The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterRow_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

8-bit unsigned single-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

8-bit unsigned three-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

8-bit unsigned four-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

8-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit unsigned single-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit unsigned three-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit unsigned four-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit single-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit three-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit four-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

16-bit four-channel 1D row convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*:

NppStatus **nppiFilterRow_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

32-bit float single-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*.

NppStatus **nppiFilterRow_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

32-bit float three-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*.

NppStatus **nppiFilterRow_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

32-bit float four-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*.

NppStatus **nppiFilterRow_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

32-bit float four-channel 1D row convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*.

NppStatus **nppiFilterRow_64f_C1R_Ctx**(const *Npp64f* *pSrc, *Npp32s* nSrcStep, *Npp64f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp64f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

64-bit float single-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow functions*.

1.10.1.7 Image Filter Row Border

1.10.1.7.1 FilterRowBorder

General purpose 1D convolution row filter with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.1.7.1.1 Common parameters for nppiFilterRowBorder functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param oSrcSize
 Source image width and height in pixels relative to pSrc.

param oSrcOffset
 The pixel offset that pSrc points to relative to the origin of the source image.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param pKernel
 Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param nMaskSize
 Width of the kernel.

param nAnchor
 X offset of the kernel origin frame of reference relative to the source pixel.

param nDivisor
 The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

param eBorderType
 The border type operation to be applied at source image border boundaries.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterRowBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus nppiFilterRowBorder_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus nppiFilterRowBorder_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel channel 8-bit unsigned 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus nppiFilterRowBorder_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned convolution 1D row filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus nppiFilterRowBorder_16u_C1R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned convolution 1D row filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus nppiFilterRowBorder_16u_C3R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus nppiFilterRowBorder_16u_C4R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel channel 16-bit 1D row unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus nppiFilterRowBorder_16u_AC4R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned 1D row convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus nppiFilterRowBorder_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus nppiFilterRowBorder_16s_C3R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus nppiFilterRowBorder_16s_C4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel channel 16-bit 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*.

NppStatus **nppiFilterRowBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit 1D row convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*:

NppStatus **nppiFilterRowBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit float 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*:

NppStatus **nppiFilterRowBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit float 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*:

NppStatus **nppiFilterRowBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit float 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*:

NppStatus **nppiFilterRowBorder_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit float 1D row convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder functions*:

1.10.1.8 Image Filter Row 32f

1.10.1.8.1 FilterRow32f

FilterRow using floating-point weights.

1.10.1.8.1.1 Common parameters for nppiFilterRow32f functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oROI
Region-Of-Interest (ROI).

param pKernel
Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param nMaskSize
Length of the linear kernel array.

param nAnchor
X offset of the kernel origin frame of reference relative to the source pixel.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterRow32f_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

8-bit unsigned single-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow32f functions*.

NppStatus **nppiFilterRow32f_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

8-bit unsigned three-channel 1D row convolution.

For common parameter descriptions, see [Common parameters for nppiFilterRow32f functions:](#).

NppStatus **nppiFilterRow32f_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

8-bit unsigned four-channel 1D row convolution.

For common parameter descriptions, see [Common parameters for nppiFilterRow32f functions:](#).

NppStatus **nppiFilterRow32f_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

8-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

For common parameter descriptions, see [Common parameters for nppiFilterRow32f functions:](#).

NppStatus **nppiFilterRow32f_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

16-bit unsigned single-channel 1D row convolution.

For common parameter descriptions, see [Common parameters for nppiFilterRow32f functions:](#).

NppStatus **nppiFilterRow32f_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

16-bit unsigned three-channel 1D row convolution.

For common parameter descriptions, see [Common parameters for nppiFilterRow32f functions:](#).

NppStatus **nppiFilterRow32f_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

16-bit unsigned four-channel 1D row convolution.

For common parameter descriptions, see [Common parameters for nppiFilterRow32f functions:](#).

NppStatus **nppiFilterRow32f_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

16-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

For common parameter descriptions, see [Common parameters for nppiFilterRow32f functions:](#).

NppStatus **nppiFilterRow32f_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

16-bit single-channel 1D row convolution.

For common parameter descriptions, see [Common parameters for nppiFilterRow32f functions:](#).

NppStatus **nppiFilterRow32f_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

16-bit three-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow32f functions*.

NppStatus **nppiFilterRow32f_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

16-bit four-channel 1D row convolution.

For common parameter descriptions, see *Common parameters for nppiFilterRow32f functions*.

NppStatus **nppiFilterRow32f_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

16-bit four-channel 1D row convolution ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRow32f functions*.

1.10.1.9 Image Filter Row Border 32f

1.10.1.9.1 FilterRowBorder32f

General purpose 1D row convolution filter using floating-point weights with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.1.9.1.1 Common parameters for nppiFilterRowBorder32f functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI*Region-Of-Interest (ROI).***param pKernel**

Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param nMaskSize

Width of the kernel.

param nAnchor

X offset of the kernel origin frame of reference relative to the source pixel.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes*

Functions

NppStatus **nppiFilterRowBorder32f_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions:*.

NppStatus **nppiFilterRowBorder32f_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions:*.

NppStatus **nppiFilterRowBorder32f_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions:*.

NppStatus **nppiFilterRowBorder32f_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned 1D row convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions*.

NppStatus **nppiFilterRowBorder32f_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions*.

NppStatus **nppiFilterRowBorder32f_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions*.

NppStatus **nppiFilterRowBorder32f_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions*.

NppStatus **nppiFilterRowBorder32f_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned 1D row convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions*.

```
NppStatus nppiFilterRowBorder32f_16s_C1R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f *pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

Single channel 16-bit 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions*.

```
NppStatus nppiFilterRowBorder32f_16s_C3R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f *pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

Three channel 16-bit 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions*.

```
NppStatus nppiFilterRowBorder32f_16s_C4R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f *pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

Four channel 16-bit 1D row convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions*.

```
NppStatus nppiFilterRowBorder32f_16s_AC4R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f *pKernel, Npp32s nMaskSize, Npp32s nAnchor, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

Four channel 16-bit 1D row convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterRowBorder32f functions*.

1.10.1.10 Image Filter 1D Window Sum

1.10.1.10.1 1D Window Sum

The set of 1D window sum functions available in the library.

1.10.1.11 Image Filter 1D Window Column Sum

1.10.1.11.1 1D Window Column Sum

1D mask Window Column Sum for 8 and 16 bit images.

1.10.1.11.1.1 Common parameters for nppiFilterSumWindowColumn functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oROI
Region-Of-Interest (ROI).

param nMaskSize
Length of the linear kernel array.

param nAnchor
Y offset of the kernel origin frame of reference relative to the source pixel.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiSumWindowColumn_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

One channel 8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 8 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn functions*:

```
NppStatus nppiSumWindowColumn_8u32f_C3R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep, Npp32f
                                                *pDst, Npp32s nDstStep, NppiSize oROI,
                                                Npp32s nMaskSize, Npp32s nAnchor,
                                                NppStreamContext nppStreamCtx)
```

Three channel 8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 3-channel 8 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn functions*:

```
NppStatus nppiSumWindowColumn_8u32f_C4R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep, Npp32f
                                                *pDst, Npp32s nDstStep, NppiSize oROI,
                                                Npp32s nMaskSize, Npp32s nAnchor,
                                                NppStreamContext nppStreamCtx)
```

Four channel 8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 8 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn functions*:

```
NppStatus nppiSumWindowColumn_16u32f_C1R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize
                                                oROI, Npp32s nMaskSize, Npp32s nAnchor,
                                                NppStreamContext nppStreamCtx)
```

One channel 16-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn functions*:

```
NppStatus nppiSumWindowColumn_16u32f_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize
                                                oROI, Npp32s nMaskSize, Npp32s nAnchor,
                                                NppStreamContext nppStreamCtx)
```

Three channel 16-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 3-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn functions*.

```
NppStatus nppiSumWindowColumn_16u32f_C4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize
                                                oROI, Npp32s nMaskSize, Npp32s nAnchor,
                                                NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn functions*.

```
NppStatus nppiSumWindowColumn_16s32f_C1R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize
                                                oROI, Npp32s nMaskSize, Npp32s nAnchor,
                                                NppStreamContext nppStreamCtx)
```

One channel 16-bit signed 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn functions*.

```
NppStatus nppiSumWindowColumn_16s32f_C3R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize
                                                oROI, Npp32s nMaskSize, Npp32s nAnchor,
                                                NppStreamContext nppStreamCtx)
```

Three channel 16-bit signed 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn functions*:

```
NppStatus nppiSumWindowColumn_16s32f_C4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize  
                                                oROI, Npp32s nMaskSize, Npp32s nAnchor,  
                                                NppStreamContext nppStreamCtx)
```

Four channel 16-bit signed 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn functions*:

1.10.1.12 Image Filter 1D Window Row Sum

1.10.1.12.1 1D Window Row Sum

1D mask Window Row Sum for 8 and 16 bit images.

1.10.1.12.1.1 Common parameters for nppiFilterSumWindowRow functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oROI

Region-Of-Interest (ROI).

param nMaskSize

Length of the linear kernel array.

param nAnchor

X offset of the kernel origin frame of reference relative to the source pixel.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiSumWindowRow_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

One channel 8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 8-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRow functions*:

NppStatus **nppiSumWindowRow_8u32f_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 8-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRow functions*:

NppStatus **nppiSumWindowRow_8u32f_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 8-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRow functions*:

NppStatus **nppiSumWindowRow_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppStreamContext* nppStreamCtx)

One channel 16-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRow functions*:

```
NppStatus nppiSumWindowRow_16u32f_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, Npp32f
                                             *pDst, Npp32s nDstStep, NppiSize oROI, Npp32s
                                             nMaskSize, Npp32s nAnchor, NppStreamContext
                                             nppStreamCtx)
```

Three channel 16-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRow functions*:

```
NppStatus nppiSumWindowRow_16u32f_C4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, Npp32f
                                             *pDst, Npp32s nDstStep, NppiSize oROI, Npp32s
                                             nMaskSize, Npp32s nAnchor, NppStreamContext
                                             nppStreamCtx)
```

Four channel 16-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRow functions*:

```
NppStatus nppiSumWindowRow_16s32f_C1R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp32f
                                             *pDst, Npp32s nDstStep, NppiSize oROI, Npp32s
                                             nMaskSize, Npp32s nAnchor, NppStreamContext
                                             nppStreamCtx)
```

One channel 16-bit signed 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRow functions*:

```
NppStatus nppiSumWindowRow_16s32f_C3R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp32f
                                             *pDst, Npp32s nDstStep, NppiSize oROI, Npp32s
                                             nMaskSize, Npp32s nAnchor, NppStreamContext
                                             nppStreamCtx)
```

Three channel 16-bit signed 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRow functions*.

```
NppStatus nppiSumWindowRow_16s32f_C4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp32f
                                             *pDst, Npp32s nDstStep, NppiSize oROI, Npp32s
                                             nMaskSize, Npp32s nAnchor, NppStreamContext
                                             nppStreamCtx)
```

Four channel 16-bit signed 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRow functions*.

1.10.1.13 Image Filter 1D Window Sum Border

1.10.1.13.1 1D Window Sum with Border Control

The set of 1D window sum functions with border control available in the library.

1.10.1.14 Image Filter 1D Window Column Sum Border

1.10.1.14.1 1D Window Column Sum Border

1D mask Window Column Sum for 8 and 16 bit images with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.1.14.1.1 Common parameters for nppiFilterSumWindowColumnBorder functions:

param pSrc*Source-Image Pointer.***param nSrcStep***Source-Image Line Step.***param oSrcSize**

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst*Destination-Image Pointer.***param nDstStep***Destination-Image Line Step.***param oROI***Region-Of-Interest (ROI).***param nMaskSize**

Length of the linear kernel array.

param nAnchor

Y offset of the kernel origin frame of reference relative to the source pixel.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes*

Functions

NppStatus **nppiSumWindowColumnBorder_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

One channel 8-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 8 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn-Border functions*.

```
NppStatus nppiSumWindowColumnBorder_8u32f_C3R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,
                                                    NppiSize oSrcSize, NppiPoint
                                                    oSrcOffset, Npp32f *pDst, Npp32s
                                                    nDstStep, NppiSize oROI, Npp32s
                                                    nMaskSize, Npp32s nAnchor,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Three channel 8-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 3-channel 8 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn-Border functions*.

```
NppStatus nppiSumWindowColumnBorder_8u32f_C4R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,
                                                    NppiSize oSrcSize, NppiPoint
                                                    oSrcOffset, Npp32f *pDst, Npp32s
                                                    nDstStep, NppiSize oROI, Npp32s
                                                    nMaskSize, Npp32s nAnchor,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Four channel 8-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 8 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn-Border functions*.

```
NppStatus nppiSumWindowColumnBorder_16u32f_C1R_Ctx(const Npp16u *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp32f *pDst,
                                                    Npp32s nDstStep, NppiSize oROI,
                                                    Npp32s nMaskSize, Npp32s nAnchor,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

One channel 16-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn-Border functions*.

```
NppStatus nppiSumWindowColumnBorder_16u32f_C3R_Ctx(const Npp16u *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp32f *pDst,
                                                    Npp32s nDstStep, NppiSize oROI,
                                                    Npp32s nMaskSize, Npp32s nAnchor,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Three channel 16-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 3-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn-Border functions*.

```
NppStatus nppiSumWindowColumnBorder_16u32f_C4R_Ctx(const Npp16u *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp32f *pDst,
                                                    Npp32s nDstStep, NppiSize oROI,
                                                    Npp32s nMaskSize, Npp32s nAnchor,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn-Border functions*.

```
NppStatus nppiSumWindowColumnBorder_16s32f_C1R_Ctx(const Npp16s *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp32f *pDst,
                                                    Npp32s nDstStep, NppiSize oROI,
                                                    Npp32s nMaskSize, Npp32s nAnchor,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

One channel 16-bit signed 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn-Border functions*:

```
NppStatus nppiSumWindowColumnBorder_16s32f_C3R_Ctx(const Npp16s *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oROI,
Npp32s nMaskSize, Npp32s nAnchor,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Three channel 16-bit signed 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn-Border functions*:

```
NppStatus nppiSumWindowColumnBorder_16s32f_C4R_Ctx(const Npp16s *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oROI,
Npp32s nMaskSize, Npp32s nAnchor,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 16-bit signed 1D (column) sum to 32f with border control.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 16 bit/pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowColumn-Border functions*:

1.10.1.15 Image Filter 1D Window Row Sum Border

1.10.1.15.1 1D Window Row Sum Border

1D mask Window Row Sum for 8 and 16 bit images with border control.

1.10.1.15.1.1 Common parameters for nppiFilterSumWindowRowBorder functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param oSrcSize
Source image width and height in pixels relative to pSrc.

param oSrcOffset
The pixel offset that pSrc points to relative to the origin of the source image.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oROI
Region-Of-Interest (ROI).

param nMaskSize
Length of the linear kernel array.

param nAnchor
X offset of the kernel origin frame of reference relative to the source pixel.

param eBorderType
The border type operation to be applied at source image border boundaries.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiSumWindowRowBorder_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oROI, *Npp32s* nMaskSize, *Npp32s* nAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

One channel 8-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 8-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRowBorder functions*.


```
NppStatus nppiSumWindowRowBorder_8u32f_C3R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep, NppiSize
oROI, Npp32s nMaskSize, Npp32s
nAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Three channel 8-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 8-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRowBorder functions*.

```
NppStatus nppiSumWindowRowBorder_8u32f_C4R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep, NppiSize
oROI, Npp32s nMaskSize, Npp32s
nAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 8-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 8-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRowBorder functions*.

```
NppStatus nppiSumWindowRowBorder_16u32f_C1R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep,
NppiSize oROI, Npp32s nMaskSize,
Npp32s nAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

One channel 16-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRowBorder functions*.

```
NppStatus nppiSumWindowRowBorder_16u32f_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                                    NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                    Npp32f *pDst, Npp32s nDstStep,  
                                                    NppiSize oROI, Npp32s nMaskSize,  
                                                    Npp32s nAnchor, NppiBorderType  
eBorderType, NppStreamContext  
nppStreamCtx)
```

Three channel 16-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRowBorder functions*.

```
NppStatus nppiSumWindowRowBorder_16u32f_C4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                                    NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                    Npp32f *pDst, Npp32s nDstStep,  
                                                    NppiSize oROI, Npp32s nMaskSize,  
                                                    Npp32s nAnchor, NppiBorderType  
eBorderType, NppStreamContext  
nppStreamCtx)
```

Four channel 16-bit unsigned 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRowBorder functions*.

```
NppStatus nppiSumWindowRowBorder_16s32f_C1R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                                    NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                    Npp32f *pDst, Npp32s nDstStep,  
                                                    NppiSize oROI, Npp32s nMaskSize,  
                                                    Npp32s nAnchor, NppiBorderType  
eBorderType, NppStreamContext  
nppStreamCtx)
```

One channel 16-bit signed 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRowBorder functions*.

```
NppStatus nppiSumWindowRowBorder_16s32f_C3R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep,
NppiSize oROI, Npp32s nMaskSize,
Npp32s nAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Three channel 16-bit signed 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRowBorder functions*.

```
NppStatus nppiSumWindowRowBorder_16s32f_C4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep,
NppiSize oROI, Npp32s nMaskSize,
Npp32s nAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Four channel 16-bit signed 1D (row) sum to 32f with border control.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 16-bit pixel input images with 32-bit floating point output.

Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

For common parameter descriptions, see *Common parameters for nppiFilterSumWindowRowBorder functions*.

1.10.2. Image Convolution

1.10.2.1 Convolution

The set convolution functions available in the library.

1.10.2.2 Image Filter

1.10.2.2.1 Filter

General purpose 2D convolution filter.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor.

1.10.2.2.1.1 Common parameters for nppiFilter functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pKernel

Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param oKernelSize

Width and Height of the rectangular kernel.

param oAnchor

X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param nDivisor

The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiFilter_8u_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus nppiFilter_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus nppiFilter_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Four channel channel 8-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus nppiFilter_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned convolution filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus nppiFilter_16u_C1R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus nppiFilter_16u_C3R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus nppiFilter_16u_C4R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Four channel channel 16-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus **nppiFilter_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned convolution filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus **nppiFilter_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus **nppiFilter_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Three channel 16-bit convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus **nppiFilter_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Four channel channel 16-bit convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus **nppiFilter_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit convolution filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus **nppiFilter_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 32-bit float convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus **nppiFilter_32f_C2R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Two channel 32-bit float convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions*.

NppStatus **nppiFilter_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 32-bit float convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions:*

NppStatus **nppiFilter_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 32-bit float convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions:*

NppStatus **nppiFilter_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 32-bit float convolution filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter functions:*

NppStatus **nppiFilter_64f_C1R_Ctx**(const *Npp64f* *pSrc, *Npp32s* nSrcStep, *Npp64f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp64f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 64-bit float convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter functions:*

1.10.2.3 Image Filter 32f

1.10.2.3.1 Filter32f

General purpose 2D convolution filter using floating point weights.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed.

1.10.2.3.1.1 Common parameters for nppiFilter32f functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pKernel

Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param oKernelSize

Width and Height of the rectangular kernel.

param oAnchor

X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilter32f_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

NppStatus **nppiFilter32f_8u_C2R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Two channel 8-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

NppStatus **nppiFilter32f_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

NppStatus **nppiFilter32f_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

NppStatus **nppiFilter32f_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned convolution filter, ignore alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

NppStatus **nppiFilter32f_8s_C1R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_8s_C2R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Two channel 8-bit signed convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_8s_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 8-bit signed convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_8s_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit signed convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_8s_AC4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit signed convolution filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_16u_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f *pKernel,  
NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned convolution filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_16s_C1R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f *pKernel,  
NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Single channel 16-bit convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_16s_C3R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f *pKernel,  
NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Three channel 16-bit convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_16s_C4R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f *pKernel,  
NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Four channel 16-bit convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_16s_AC4R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f *pKernel,  
NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Four channel 16-bit convolution filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_32s_C1R_Ctx(const Npp32s *pSrc, int nSrcStep, Npp32s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f *pKernel,  
NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Single channel 32-bit convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_32s_C3R_Ctx(const Npp32s *pSrc, int nSrcStep, Npp32s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f *pKernel,  
NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Three channel 32-bit convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

NppStatus **nppiFilter32f_32s_C4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 32-bit convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_32s_AC4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 32-bit convolution filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_16f_C1R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit floating point convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_16f_C3R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 16-bit floating point convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_16f_C4R_Ctx**(const *Npp16f* *pSrc, int nSrcStep, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit floating point convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_8u16s_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned to 16-bit signed convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions*.

NppStatus **nppiFilter32f_8u16s_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned to 16-bit signed convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_8u16s_AC4R_Ctx(const Npp8u *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f  
*pKernel, NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Four channel 8-bit unsigned to 16-bit signed convolution filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_8s16s_C1R_Ctx(const Npp8s *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f *pKernel,  
NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Single channel 8-bit to 16-bit signed convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_8s16s_C3R_Ctx(const Npp8s *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f *pKernel,  
NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Three channel 8-bit to 16-bit signed convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_8s16s_C4R_Ctx(const Npp8s *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f *pKernel,  
NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Four channel 8-bit to 16-bit signed convolution filter.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

```
NppStatus nppiFilter32f_8s16s_AC4R_Ctx(const Npp8s *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f  
*pKernel, NppiSize oKernelSize, NppiPoint oAnchor,  
NppStreamContext nppStreamCtx)
```

Four channel 8-bit to 16-bit signed convolution filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilter32f functions:*.

1.10.2.4 Image Filter Border

1.10.2.4.1 FilterBorder

General purpose 2D convolution filter with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.2.4.1.1 Common parameters for nppiFilterBorder functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param oSrcSize
 Source image width and height in pixels relative to pSrc.

param oSrcOffset
 The pixel offset that pSrc points to relative to the origin of the source image.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param pKernel
 Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param oKernelSize
 Width and Height of the rectangular kernel.

param oAnchor
 X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param nDivisor
 The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

param eBorderType
 The border type operation to be applied at source image border boundaries.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions:*.

NppStatus nppiFilterBorder_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

NppStatus nppiFilterBorder_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel channel 8-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

NppStatus nppiFilterBorder_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

NppStatus nppiFilterBorder_16u_C1R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

NppStatus nppiFilterBorder_16u_C3R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

NppStatus nppiFilterBorder_16u_C4R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *Npp32s* nDivisor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel channel 16-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

```
NppStatus nppiFilterBorder_16u_AC4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst,
Npp32s nDstStep, NppiSize oSizeROI, const
Npp32s *pKernel, NppiSize oKernelSize, NppiPoint
oAnchor, Npp32s nDivisor, NppiBorderType
eBorderType, NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

```
NppStatus nppiFilterBorder_16s_C1R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
Npp32s nDstStep, NppiSize oSizeROI, const Npp32s
*pKernel, NppiSize oKernelSize, NppiPoint oAnchor,
Npp32s nDivisor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Single channel 16-bit convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

```
NppStatus nppiFilterBorder_16s_C3R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
Npp32s nDstStep, NppiSize oSizeROI, const Npp32s
*pKernel, NppiSize oKernelSize, NppiPoint oAnchor,
Npp32s nDivisor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Three channel 16-bit convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

```
NppStatus nppiFilterBorder_16s_C4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
Npp32s nDstStep, NppiSize oSizeROI, const Npp32s
*pKernel, NppiSize oKernelSize, NppiPoint oAnchor,
Npp32s nDivisor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel channel 16-bit convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

```
NppStatus nppiFilterBorder_16s_AC4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
Npp32s nDstStep, NppiSize oSizeROI, const
Npp32s *pKernel, NppiSize oKernelSize, NppiPoint
oAnchor, Npp32s nDivisor, NppiBorderType
eBorderType, NppStreamContext nppStreamCtx)
```

Four channel 16-bit convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

```
NppStatus nppiFilterBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI, const Npp32f
*pKernel, NppiSize oKernelSize, NppiPoint oAnchor,
NppiBorderType eBorderType, NppStreamContext
nppStreamCtx)
```

Single channel 32-bit float convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

```
NppStatus nppiFilterBorder_32f_C2R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
                                         oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                         Npp32s nDstStep, NppiSize oSizeROI, const Npp32f
                                         *pKernel, NppiSize oKernelSize, NppiPoint oAnchor,
                                         NppiBorderType eBorderType, NppStreamContext
                                         nppStreamCtx)
```

Two channel 32-bit float convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

```
NppStatus nppiFilterBorder_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
                                         oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                         Npp32s nDstStep, NppiSize oSizeROI, const Npp32f
                                         *pKernel, NppiSize oKernelSize, NppiPoint oAnchor,
                                         NppiBorderType eBorderType, NppStreamContext
                                         nppStreamCtx)
```

Three channel 32-bit float convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

```
NppStatus nppiFilterBorder_32f_C4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
                                         oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                         Npp32s nDstStep, NppiSize oSizeROI, const Npp32f
                                         *pKernel, NppiSize oKernelSize, NppiPoint oAnchor,
                                         NppiBorderType eBorderType, NppStreamContext
                                         nppStreamCtx)
```

Four channel 32-bit float convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

```
NppStatus nppiFilterBorder_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
                                         oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                         Npp32s nDstStep, NppiSize oSizeROI, const
                                         Npp32f *pKernel, NppiSize oKernelSize, NppiPoint
                                         oAnchor, NppiBorderType eBorderType,
                                         NppStreamContext nppStreamCtx)
```

Four channel 32-bit float convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder functions*:

1.10.2.5 Image Filter Border 32f

1.10.2.5.1 FilterBorder32f

General purpose 2D convolution filter using floating-point weights with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.2.5.1.1 Common parameters for nppiFilterBorder32f functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param oSrcSize
 Source image width and height in pixels relative to pSrc.

param oSrcOffset
 The pixel offset that pSrc points to relative to the origin of the source image.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param pKernel
 Device memory pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

param oKernelSize
 Width and Height of the rectangular kernel.

param oAnchor
 X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param eBorderType
 The border type operation to be applied at source image border boundaries.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterBorder32f_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_8u_C2R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Two channel 8-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_8u_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned convolution filter with border control, ignore alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_8s_C1R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_8s_C2R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Two channel 8-bit signed convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_8s_C3R_Ctx(const Npp8s *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp8s *pDst, int
nDstStep, NppiSize oSizeROI, const Npp32f
*pKernel, NppiSize oKernelSize, NppiPoint
oAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Three channel 8-bit signed convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_8s_C4R_Ctx(const Npp8s *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp8s *pDst, int
nDstStep, NppiSize oSizeROI, const Npp32f
*pKernel, NppiSize oKernelSize, NppiPoint
oAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 8-bit signed convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_8s_AC4R_Ctx(const Npp8s *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp8s *pDst, int
nDstStep, NppiSize oSizeROI, const Npp32f
*pKernel, NppiSize oKernelSize, NppiPoint
oAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 8-bit signed convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_16u_C1R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst,
int nDstStep, NppiSize oSizeROI, const Npp32f
*pKernel, NppiSize oKernelSize, NppiPoint
oAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Single channel 16-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_16u_C3R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst,
int nDstStep, NppiSize oSizeROI, const Npp32f
*pKernel, NppiSize oKernelSize, NppiPoint
oAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Three channel 16-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_16u_C4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst,
int nDstStep, NppiSize oSizeROI, const Npp32f
*pKernel, NppiSize oKernelSize, NppiPoint
oAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_16u_AC4R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiSize
                                             oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst,
                                             int nDstStep, NppiSize oSizeROI, const Npp32f
                                             *pKernel, NppiSize oKernelSize, NppiPoint
                                             oAnchor, NppiBorderType eBorderType,
                                             NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_16s_C1R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize
                                             oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
                                             int nDstStep, NppiSize oSizeROI, const Npp32f
                                             *pKernel, NppiSize oKernelSize, NppiPoint
                                             oAnchor, NppiBorderType eBorderType,
                                             NppStreamContext nppStreamCtx)
```

Single channel 16-bit convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_16s_C3R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize
                                             oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
                                             int nDstStep, NppiSize oSizeROI, const Npp32f
                                             *pKernel, NppiSize oKernelSize, NppiPoint
                                             oAnchor, NppiBorderType eBorderType,
                                             NppStreamContext nppStreamCtx)
```

Three channel 16-bit convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_16s_C4R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize
                                             oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
                                             int nDstStep, NppiSize oSizeROI, const Npp32f
                                             *pKernel, NppiSize oKernelSize, NppiPoint
                                             oAnchor, NppiBorderType eBorderType,
                                             NppStreamContext nppStreamCtx)
```

Four channel 16-bit convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_16s_AC4R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize
                                             oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
                                             int nDstStep, NppiSize oSizeROI, const Npp32f
                                             *pKernel, NppiSize oKernelSize, NppiPoint
                                             oAnchor, NppiBorderType eBorderType,
                                             NppStreamContext nppStreamCtx)
```

Four channel 16-bit convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

NppStatus nppiFilterBorder32f_32s_C1R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_32s_C3R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_32s_C4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_32s_AC4R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_16f_C1R_Ctx(const *Npp16f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit floating point convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus nppiFilterBorder32f_16f_C3R_Ctx(const *Npp16f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit floating point convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_16f_C4R_Ctx(const Npp16f *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16f *pDst,
int nDstStep, NppiSize oSizeROI, const Npp32f
*pKernel, NppiSize oKernelSize, NppiPoint
oAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 16-bit floating point convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_8u16s_C1R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
int nDstStep, NppiSize oSizeROI, const
Npp32f *pKernel, NppiSize oKernelSize,
NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Single channel 8-bit unsigned to 16-bit signed convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_8u16s_C3R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
int nDstStep, NppiSize oSizeROI, const
Npp32f *pKernel, NppiSize oKernelSize,
NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Three channel 8-bit unsigned to 16-bit signed convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_8u16s_C4R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16s *pDst,
int nDstStep, NppiSize oSizeROI, const
Npp32f *pKernel, NppiSize oKernelSize,
NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Four channel 8-bit unsigned to 16-bit signed convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

```
NppStatus nppiFilterBorder32f_8u16s_AC4R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16s
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp32f *pKernel, NppiSize
oKernelSize, NppiPoint oAnchor,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 8-bit unsigned to 16-bit signed convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*:

NppStatus **nppiFilterBorder32f_8s16s_C1R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit to 16-bit signed convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus **nppiFilterBorder32f_8s16s_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit to 16-bit signed convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus **nppiFilterBorder32f_8s16s_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit to 16-bit signed convolution filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

NppStatus **nppiFilterBorder32f_8s16s_AC4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pKernel, *NppiSize* oKernelSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit to 16-bit signed convolution filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBorder32f functions*.

1.10.3. 2D Fixed Linear Filters

1.10.3.1 2D Fixed Linear Filters

The set of 2D fixed linear filtering functions available in the library.

1.10.3.2 Image Filter Box

1.10.3.2.1 FilterBox

Computes the average pixel values of the pixels under a rectangular mask.

1.10.3.2.1.1 Common parameters for nppiFilterBox functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param oMaskSize
Width and Height of the neighborhood region for the local Avg operation.

param oAnchor
X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterBox_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst,
Npp32s nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize,
NppiPoint oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions:*.

NppStatus **nppiFilterBox_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst,
Npp32s nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize,
NppiPoint oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions:*.

NppStatus **nppiFilterBox_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst,
Npp32s nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize,
NppiPoint oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*:

NppStatus **nppiFilterBox_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned box filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*:

NppStatus **nppiFilterBox_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*:

NppStatus **nppiFilterBox_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*:

NppStatus **nppiFilterBox_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*:

NppStatus **nppiFilterBox_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned box filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*:

NppStatus **nppiFilterBox_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*:

NppStatus **nppiFilterBox_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 16-bit box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*:

NppStatus **nppiFilterBox_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*.

NppStatus **nppiFilterBox_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit box filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*.

NppStatus **nppiFilterBox_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*.

NppStatus **nppiFilterBox_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*.

NppStatus **nppiFilterBox_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*.

NppStatus **nppiFilterBox_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point box filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*.

NppStatus **nppiFilterBox_64f_C1R_Ctx**(const *Npp64f* *pSrc, *Npp32s* nSrcStep, *Npp64f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 64-bit floating-point box filter.

For common parameter descriptions, see *Common parameters for nppiFilterBox functions*.

1.10.3.3 Image Filter Box Border

1.10.3.3.1 FilterBoxBorder

Computes the average pixel values of the pixels under a rectangular mask with border control. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported. *

1.10.3.3.1.1 Common parameters for nppiFilterBoxBorder functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param oSrcSize
 Source image width and height in pixels relative to pSrc.

param oSrcOffset
 The pixel offset that pSrc points to relative to the origin of the source image.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param oMaskSize
 Width and Height of the neighborhood region for the local Avg operation.

param oAnchor
 X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param eBorderType
 The border type operation to be applied at source image border boundaries.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterBoxBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus **nppiFilterBoxBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus **nppiFilterBoxBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus **nppiFilterBoxBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned box filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus **nppiFilterBoxBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus **nppiFilterBoxBorder_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus nppiFilterBoxBorder_16u_C4R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus nppiFilterBoxBorder_16u_AC4R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned box filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus nppiFilterBoxBorder_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus nppiFilterBoxBorder_16s_C3R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus nppiFilterBoxBorder_16s_C4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*.

NppStatus nppiFilterBoxBorder_16s_AC4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit box filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*:

```
NppStatus nppiFilterBoxBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI, NppiSize
oMaskSize, NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Single channel 32-bit floating-point box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*:

```
NppStatus nppiFilterBoxBorder_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI, NppiSize
oMaskSize, NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Three channel 32-bit floating-point box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*:

```
NppStatus nppiFilterBoxBorder_32f_C4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI, NppiSize
oMaskSize, NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Four channel 32-bit floating-point box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*:

```
NppStatus nppiFilterBoxBorder_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 32-bit floating-point box filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorder functions*:

1.10.3.4 Image Filter Box Border Advanced

1.10.3.4.1 FilterBoxBorderAdvanced

Computes the average pixel values of the pixels under a rectangular mask with border control. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image. These versions of the functions are intended to significantly improve performance when very large mask sizes are used.

Currently only the NPP_BORDER_REPLICATE border type operation is supported. *

1.10.3.4.1.1 Common parameters for nppiFilterBoxBorderAdvanced functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param oSrcSize
 Source image width and height in pixels relative to pSrc.

param oSrcOffset
 The pixel offset that pSrc points to relative to the origin of the source image.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param oMaskSize
 Width and Height of the neighborhood region for the local Avg operation.

param oAnchor
 X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param eBorderType
 The border type operation to be applied at source image border boundaries.

param pBuffer
 Pointer to device memory buffer of size hpBufferSize bytes returned by calling nppiFilterBoxBorderAdvancedGetDeviceBufferSize.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterBoxBorderAdvancedGetDeviceBufferSize**(*NppiSize* oSizeROI, int nChannels, int *hpBufferSize)

Returns the required size of host memory buffer needed by most nppiFilterBoxBorderAdvanced functions.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **nChannels** – The number of channels in the image to be filtered (1, 3, or 4).
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
Scratch Buffer and Host Pointer.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiFilterBoxBorderAdvancedGetDeviceBufferSize_64**(*NppiSize* oSizeROI, int nChannels, int *hpBufferSize)

Returns the required size of host memory buffer needed by nppiFilterBoxBorderAdvanced functions with 64-bit image data.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nChannels** – The number of channels in the image to be filtered (1, 3, or 4).
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiFilterBoxBorderAdvanced_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorderAdvanced functions*:

NppStatus **nppiFilterBoxBorderAdvanced_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorderAdvanced functions*:

NppStatus **nppiFilterBoxBorderAdvanced_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorderAdvanced functions*.

```
NppStatus nppiFilterBoxBorderAdvanced_16u_C1R_Ctx(const Npp16u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16u *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType
eBorderType, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Single channel 16-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorderAdvanced functions*.

```
NppStatus nppiFilterBoxBorderAdvanced_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16u *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType
eBorderType, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Three channel 16-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorderAdvanced functions*.

```
NppStatus nppiFilterBoxBorderAdvanced_16u_C4R_Ctx(const Npp16u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16u *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType
eBorderType, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorderAdvanced functions*.

```
NppStatus nppiFilterBoxBorderAdvanced_16s_C1R_Ctx(const Npp16s *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16s *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType
eBorderType, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Single channel 16-bit box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorderAdvanced functions*.

```
NppStatus nppiFilterBoxBorderAdvanced_16s_C3R_Ctx(const Npp16s *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16s *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType
eBorderType, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Three channel 16-bit box filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterBoxBorderAdvanced functions](#):

```
NppStatus nppiFilterBoxBorderAdvanced_16s_C4R_Ctx(const Npp16s *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16s *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType
eBorderType, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Four channel 16-bit box filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterBoxBorderAdvanced functions](#):

```
NppStatus nppiFilterBoxBorderAdvanced_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType
eBorderType, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Single channel 32-bit floating-point box filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterBoxBorderAdvanced functions](#):

```
NppStatus nppiFilterBoxBorderAdvanced_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType
eBorderType, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Three channel 32-bit floating-point box filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterBoxBorderAdvanced functions](#):

```
NppStatus nppiFilterBoxBorderAdvanced_32f_C4R_Ctx(const Npp32f *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType
eBorderType, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Four channel 32-bit floating-point box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorderAdvanced functions*:

```
NppStatus nppiFilterBoxBorderAdvanced_64f_C1R_Ctx(const Npp64f *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp64f *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType
eBorderType, Npp8u *pBuffer64,
NppStreamContext nppStreamCtx)
```

Single channel 64-bit floating-point box filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBoxBorderAdvanced functions*:

1.10.3.5 Image Filter Threshold Adaptive Box Border

1.10.3.5.1 FilterThresholdAdaptiveBoxBorder

Computes the average pixel values of the pixels under a square mask with border control. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image. Once the neighborhood average around a source pixel is determined the source pixel is compared to the average - nDelta and if the source pixel is greater than that average the corresponding destination pixel is set to nValGT, otherwise nValLE.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI*Region-Of-Interest (ROI).***param oMaskSize**

Width and Height of the neighborhood region for the local Avg operation, Width and Height must be equal and odd.

param nDelta

Neighborhood average adjustment value

param nValGT

Destination output value if source pixel is greater than average.

param nValLE

Destination output value if source pixel is less than or equal to average.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes*

Functions

```
NppStatus nppiFilterThresholdAdaptiveBoxBorder_8u_C1R_Ctx(const Npp8u *pSrc, Npp32s
    nSrcStep, NppiSize
    oSrcSize, NppiPoint
    oSrcOffset, Npp8u *pDst,
    Npp32s nDstStep, NppiSize
    oSizeROI, NppiSize
    oMaskSize, Npp32f nDelta,
    Npp8u nValGT, Npp8u
    nValLE, NppiBorderType
    eBorderType,
    NppStreamContext
    nppStreamCtx)
```

Single channel 8-bit unsigned threshold adaptive box filter with border control.

1.10.4. Rank Filters

1.10.4.1 Rank Filters

The set of functions providing min/max/median values for rectangular mask region with/without border available in the library.

1.10.4.2 Image Filter Max

1.10.4.2.1 FilterMax

Result pixel value is the maximum of pixel values under the rectangular mask region.

1.10.4.2.1.1 Common parameters for nppiFilterMax functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step plus mask size width.

param oSizeROI
Region-Of-Interest (ROI).

param oMaskSize
Width and Height of the neighborhood region for the local Max operation.

param oAnchor
X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterMax_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions:*.

NppStatus **nppiFilterMax_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions:*.

NppStatus **nppiFilterMax_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*:

NppStatus **nppiFilterMax_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst,
Npp32s nDstStep, *NppiSize* oSizeROI, *NppiSize*
oMaskSize, *NppiPoint* oAnchor, *NppStreamContext*
nppStreamCtx)

Four channel 8-bit unsigned maximum filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*:

NppStatus **nppiFilterMax_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst,
Npp32s nDstStep, *NppiSize* oSizeROI, *NppiSize*
oMaskSize, *NppiPoint* oAnchor, *NppStreamContext*
nppStreamCtx)

Single channel 16-bit unsigned maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*:

NppStatus **nppiFilterMax_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst,
Npp32s nDstStep, *NppiSize* oSizeROI, *NppiSize*
oMaskSize, *NppiPoint* oAnchor, *NppStreamContext*
nppStreamCtx)

Three channel 16-bit unsigned maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*:

NppStatus **nppiFilterMax_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst,
Npp32s nDstStep, *NppiSize* oSizeROI, *NppiSize*
oMaskSize, *NppiPoint* oAnchor, *NppStreamContext*
nppStreamCtx)

Four channel 16-bit unsigned maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*:

NppStatus **nppiFilterMax_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst,
Npp32s nDstStep, *NppiSize* oSizeROI, *NppiSize*
oMaskSize, *NppiPoint* oAnchor, *NppStreamContext*
nppStreamCtx)

Four channel 16-bit unsigned maximum filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*:

NppStatus **nppiFilterMax_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst,
Npp32s nDstStep, *NppiSize* oSizeROI, *NppiSize*
oMaskSize, *NppiPoint* oAnchor, *NppStreamContext*
nppStreamCtx)

Single channel 16-bit signed maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*:

NppStatus **nppiFilterMax_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst,
Npp32s nDstStep, *NppiSize* oSizeROI, *NppiSize*
oMaskSize, *NppiPoint* oAnchor, *NppStreamContext*
nppStreamCtx)

Three channel 16-bit signed maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*:

NppStatus **nppiFilterMax_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*.

NppStatus **nppiFilterMax_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed maximum filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*.

NppStatus **nppiFilterMax_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*.

NppStatus **nppiFilterMax_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*.

NppStatus **nppiFilterMax_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point maximum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*.

NppStatus **nppiFilterMax_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point maximum filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMax functions*.

1.10.4.3 Image Filter Max Border

1.10.4.3.1 FilterMaxBorder

Result pixel value is the maximum of pixel values under the rectangular mask region with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.4.3.1.1 Common parameters for nppiFilterMaxBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param oMaskSize

Width and Height of the neighborhood region for the local Max operation.

param oAnchor

X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiFilterMaxBorder_8u_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*:

NppStatus nppiFilterMaxBorder_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*:

NppStatus nppiFilterMaxBorder_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*:

NppStatus nppiFilterMaxBorder_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned maximum filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*:

NppStatus nppiFilterMaxBorder_16u_C1R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*:

NppStatus nppiFilterMaxBorder_16u_C3R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*:

NppStatus **nppiFilterMaxBorder_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*.

NppStatus **nppiFilterMaxBorder_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned maximum filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*.

NppStatus **nppiFilterMaxBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*.

NppStatus **nppiFilterMaxBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*.

NppStatus **nppiFilterMaxBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*.

NppStatus **nppiFilterMaxBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed maximum filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*.

```
NppStatus nppiFilterMaxBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI, NppiSize
oMaskSize, NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Single channel 32-bit floating-point maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*.

```
NppStatus nppiFilterMaxBorder_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI, NppiSize
oMaskSize, NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Three channel 32-bit floating-point maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*.

```
NppStatus nppiFilterMaxBorder_32f_C4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI, NppiSize
oMaskSize, NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Four channel 32-bit floating-point maximum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*.

```
NppStatus nppiFilterMaxBorder_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 32-bit floating-point maximum filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMaxBorder functions*.

1.10.4.4 Image Filter Min

1.10.4.4.1 FilterMin

Result pixel value is the minimum of pixel values under the rectangular mask region.

1.10.4.4.1.1 Common parameters for nppiFilterMin functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step plus mask size width.

param oSizeROI
Region-Of-Interest (ROI).

param oMaskSize
Width and Height of the neighborhood region for the local Min operation.

param oAnchor
X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterMin_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*.

NppStatus **nppiFilterMin_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*.

NppStatus **nppiFilterMin_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*.

NppStatus **nppiFilterMin_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned minimum filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*:

NppStatus **nppiFilterMin_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*:

NppStatus **nppiFilterMin_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*:

NppStatus **nppiFilterMin_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*:

NppStatus **nppiFilterMin_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned minimum filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*:

NppStatus **nppiFilterMin_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*:

NppStatus **nppiFilterMin_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*:

NppStatus **nppiFilterMin_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*:

NppStatus **nppiFilterMin_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed minimum filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*.

NppStatus **nppiFilterMin_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*.

NppStatus **nppiFilterMin_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*.

NppStatus **nppiFilterMin_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point minimum filter.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*.

NppStatus **nppiFilterMin_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point minimum filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMin functions*.

1.10.4.5 Image Filter Min Border

1.10.4.5.1 FilterMinBorder

Result pixel value is the minimum of pixel values under the rectangular mask region with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.4.5.1.1 Common parameters for nppiFilterMinBorder functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param oSrcSize
 Source image width and height in pixels relative to pSrc.

param oSrcOffset
 The pixel offset that pSrc points to relative to the origin of the source image.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param oMaskSize
 Width and Height of the neighborhood region for the local Min operation.

param oAnchor
 X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param eBorderType
 The border type operation to be applied at source image border boundaries.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterMinBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions:*.

NppStatus **nppiFilterMinBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions:*.

NppStatus **nppiFilterMinBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus **nppiFilterMinBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned minimum filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus **nppiFilterMinBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus **nppiFilterMinBorder_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus **nppiFilterMinBorder_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus **nppiFilterMinBorder_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned minimum filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus nppiFilterMinBorder_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus nppiFilterMinBorder_16s_C3R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus nppiFilterMinBorder_16s_C4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus nppiFilterMinBorder_16s_AC4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed minimum filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus nppiFilterMinBorder_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*.

NppStatus nppiFilterMinBorder_32f_C3R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*:

```
NppStatus nppiFilterMinBorder_32f_C4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI, NppiSize
oMaskSize, NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Four channel 32-bit floating-point minimum filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*:

```
NppStatus nppiFilterMinBorder_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiSize oMaskSize, NppiPoint
oAnchor, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 32-bit floating-point minimum filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMinBorder functions*:

1.10.4.6 Image Filter Median

1.10.4.6.1 FilterMedian

Result pixel value is the median of pixel values under the rectangular mask region.

1.10.4.6.1.1 Common parameters for nppiFilterMedian functions:

1.10.4.6.1.2 Common parameters for nppiFilterMedianGetBufferSize functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param oMaskSize

Width and Height of the neighborhood region for the local Median operation.

param oAnchor

X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param pBuffer

Device memory pointer to the user-allocated scratch buffer required for the Median operation.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

param oSizeROI

Region-Of-Interest (ROI).

param oMaskSize

Width and Height of the neighborhood region for the local Median operation.

param nBufferSize

Pointer to the size of the scratch buffer required for the Median operation.

return

Image Data Related Error Codes

Functions

NppStatus **nppiFilterMedian_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions:*.

NppStatus **nppiFilterMedian_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions:*.

NppStatus **nppiFilterMedian_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions:*.

NppStatus **nppiFilterMedian_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions:*.

```
NppStatus nppiFilterMedian_16u_C1R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, Npp16u
                                         *pDst, Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiSize oMaskSize, NppiPoint oAnchor, Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Single channel 16-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedian_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, Npp16u
                                         *pDst, Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiSize oMaskSize, NppiPoint oAnchor, Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Three channel 16-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedian_16u_C4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, Npp16u
                                         *pDst, Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiSize oMaskSize, NppiPoint oAnchor, Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedian_16u_AC4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, Npp16u
                                         *pDst, Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiSize oMaskSize, NppiPoint oAnchor, Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedian_16s_C1R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp16s
                                         *pDst, Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiSize oMaskSize, NppiPoint oAnchor, Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Single channel 16-bit signed median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedian_16s_C3R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp16s
                                         *pDst, Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiSize oMaskSize, NppiPoint oAnchor, Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Three channel 16-bit signed median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedian_16s_C4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp16s
                                         *pDst, Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiSize oMaskSize, NppiPoint oAnchor, Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Four channel 16-bit signed median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedian_16s_AC4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep, Npp16s
                                         *pDst, Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiSize oMaskSize, NppiPoint oAnchor, Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Four channel 16-bit signed median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions:*.

NppStatus **nppiFilterMedian_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions:*.

NppStatus **nppiFilterMedian_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions:*.

NppStatus **nppiFilterMedian_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions:*.

NppStatus **nppiFilterMedian_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions:*.

NppStatus **nppiFilterMedianGetBufferSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions:*.

NppStatus **nppiFilterMedianGetBufferSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions:*.

NppStatus **nppiFilterMedianGetBufferSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions:*.

NppStatus **nppiFilterMedianGetBufferSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus **nppiFilterMedianGetBufferSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus **nppiFilterMedianGetBufferSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus **nppiFilterMedianGetBufferSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus **nppiFilterMedianGetBufferSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus **nppiFilterMedianGetBufferSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus **nppiFilterMedianGetBufferSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus nppiFilterMedianGetBufferSize_16s_C4R_Ctx(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus nppiFilterMedianGetBufferSize_16s_AC4R_Ctx(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus nppiFilterMedianGetBufferSize_32f_C1R_Ctx(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus nppiFilterMedianGetBufferSize_32f_C3R_Ctx(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus nppiFilterMedianGetBufferSize_32f_C4R_Ctx(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

NppStatus nppiFilterMedianGetBufferSize_32f_AC4R_Ctx(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedianGetBufferSize functions*.

1.10.4.7 Image Filter Median Border

1.10.4.7.1 FilterMedianBorder

Result pixel value is the median of pixel values under the rectangular mask region.

1.10.4.7.1.1 Common parameters for nppiFilterMedianBorder functions:

1.10.4.7.1.2 Common parameters for nppiFilterMedianBorderGetBufferSize functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param oMaskSize

Width and Height of the neighborhood region for the local Median operation.

param oAnchor

X and Y offsets of the kernel origin frame of reference relative to the source pixel.

param pBuffer

Device memory pointer to the user-allocated scratch buffer required for the Median operation.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

param oSizeROI

Region-Of-Interest (ROI).

param oMaskSize

Width and Height of the neighborhood region for the local Median operation.

param nBufferSize

Pointer to the size of the scratch buffer required for the Median operation.

return

Image Data Related Error Codes

Functions

NppStatus **nppiFilterMedianBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*:

NppStatus **nppiFilterMedianBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*:

NppStatus **nppiFilterMedianBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*:

NppStatus **nppiFilterMedianBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*:

NppStatus **nppiFilterMedianBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*:

```
NppStatus nppiFilterMedianBorder_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,
                                              Npp16u *pDst, Npp32s nDstStep, NppiSize
                                              oSizeROI, NppiSize oMaskSize, NppiPoint
                                              oAnchor, Npp8u *pBuffer, NppiBorderType
                                              eBorderType, NppStreamContext
                                              nppStreamCtx)
```

Three channel 16-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*:

```
NppStatus nppiFilterMedianBorder_16u_C4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,
                                              Npp16u *pDst, Npp32s nDstStep, NppiSize
                                              oSizeROI, NppiSize oMaskSize, NppiPoint
                                              oAnchor, Npp8u *pBuffer, NppiBorderType
                                              eBorderType, NppStreamContext
                                              nppStreamCtx)
```

Four channel 16-bit unsigned median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*:

```
NppStatus nppiFilterMedianBorder_16u_AC4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,
                                              Npp16u *pDst, Npp32s nDstStep, NppiSize
                                              oSizeROI, NppiSize oMaskSize, NppiPoint
                                              oAnchor, Npp8u *pBuffer, NppiBorderType
                                              eBorderType, NppStreamContext
                                              nppStreamCtx)
```

Four channel 16-bit unsigned median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*:

```
NppStatus nppiFilterMedianBorder_16s_C1R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,
                                              Npp16s *pDst, Npp32s nDstStep, NppiSize
                                              oSizeROI, NppiSize oMaskSize, NppiPoint
                                              oAnchor, Npp8u *pBuffer, NppiBorderType
                                              eBorderType, NppStreamContext
                                              nppStreamCtx)
```

Single channel 16-bit signed median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*:

```
NppStatus nppiFilterMedianBorder_16s_C3R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,
                                              Npp16s *pDst, Npp32s nDstStep, NppiSize
                                              oSizeROI, NppiSize oMaskSize, NppiPoint
                                              oAnchor, Npp8u *pBuffer, NppiBorderType
                                              eBorderType, NppStreamContext
                                              nppStreamCtx)
```

Three channel 16-bit signed median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*:

```
NppStatus nppiFilterMedianBorder_16s_C4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp16s *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiSize oMaskSize, NppiPoint
oAnchor, Npp8u *pBuffer, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Four channel 16-bit signed median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedianBorder_16s_AC4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp16s *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiSize oMaskSize, NppiPoint
oAnchor, Npp8u *pBuffer, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Four channel 16-bit signed median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedianBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiSize oMaskSize, NppiPoint
oAnchor, Npp8u *pBuffer, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Single channel 32-bit floating-point median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedianBorder_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiSize oMaskSize, NppiPoint
oAnchor, Npp8u *pBuffer, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Three channel 32-bit floating-point median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

```
NppStatus nppiFilterMedianBorder_32f_C4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiSize oMaskSize, NppiPoint
oAnchor, Npp8u *pBuffer, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Four channel 32-bit floating-point median filter.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

NppStatus **nppiFilterMedianBorder_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedian functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-Buffer-Size functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-Buffer-Size functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-Buffer-Size functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-Buffer-Size functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-BufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-BufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-BufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-BufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-BufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-BufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-BufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-BufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGet-BufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGetBufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point median filter scratch memory size.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGetBufferSize functions*.

NppStatus **nppiFilterMedianBorderGetBufferSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, *NppiSize* oMaskSize, *Npp32u* *nBufferSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point median filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterMedianBorderGetBufferSize functions*.

1.10.5. Fixed Filters

1.10.5.1 Fixed Filters

Fixed filters perform linear filtering operations (such as convolutions) with predefined kernels of fixed sizes. Note that this section also contains a few dynamic kernel filters, namely GaussAdvanced and Bilateral.

Some of the fixed filters have versions with border control. For these functions, if any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported for these functions.

1.10.5.2 Image Filter Prewitt

1.10.5.2.1 FilterPrewitt

Filters the image using a Prewitt filter kernel.

1.10.5.2.1.1 Common parameters for nppiFilterPrewitt functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

FilterPrewittHoriz

Filters the image using a horizontal Prewitt filter kernel:

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

NppStatus **nppiFilterPrewittHoriz_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned horizontal Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*:

NppStatus **nppiFilterPrewittHoriz_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned horizontal Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*:

NppStatus nppiFilterPrewittHoriz_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus nppiFilterPrewittHoriz_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Prewitt filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus nppiFilterPrewittHoriz_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed horizontal Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus nppiFilterPrewittHoriz_16s_C3R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed horizontal Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus nppiFilterPrewittHoriz_16s_C4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed horizontal Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus nppiFilterPrewittHoriz_16s_AC4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed horizontal Prewitt filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus nppiFilterPrewittHoriz_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point horizontal Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus nppiFilterPrewittHoriz_32f_C3R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point horizontal Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*:

NppStatus **nppiFilterPrewittHoriz_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point horizontal Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*:

NppStatus **nppiFilterPrewittHoriz_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point horizontal Prewitt filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*:

FilterPrewittVert

Filters the image using a vertical Prewitt filter kernel:

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

NppStatus **nppiFilterPrewittVert_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned vertical Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*:

NppStatus **nppiFilterPrewittVert_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned vertical Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*:

NppStatus **nppiFilterPrewittVert_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned vertical Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*:

NppStatus **nppiFilterPrewittVert_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned vertical Prewitt filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus **nppiFilterPrewittVert_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep,
Npp16s *pDst, *Npp32s* nDstStep, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed vertical Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus **nppiFilterPrewittVert_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep,
Npp16s *pDst, *Npp32s* nDstStep, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed vertical Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus **nppiFilterPrewittVert_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep,
Npp16s *pDst, *Npp32s* nDstStep, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed vertical Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus **nppiFilterPrewittVert_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep,
Npp16s *pDst, *Npp32s* nDstStep, *NppiSize*
oSizeROI, *NppStreamContext*
nppStreamCtx)

Four channel 16-bit signed vertical Prewitt filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus **nppiFilterPrewittVert_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep,
Npp32f *pDst, *Npp32s* nDstStep, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point vertical Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus **nppiFilterPrewittVert_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep,
Npp32f *pDst, *Npp32s* nDstStep, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point vertical Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus **nppiFilterPrewittVert_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep,
Npp32f *pDst, *Npp32s* nDstStep, *NppiSize*
oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point vertical Prewitt filter.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

NppStatus **nppiFilterPrewittVert_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep,
Npp32f *pDst, *Npp32s* nDstStep, *NppiSize*
oSizeROI, *NppStreamContext*
nppStreamCtx)

Four channel 32-bit floating-point vertical Prewitt filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewitt functions*.

1.10.5.3 Image Filter Prewitt Border

1.10.5.3.1 FilterPrewittBorder

Filters the image using a Prewitt filter kernel with border control.

1.10.5.3.1.1 Common parameters for nppiFilterPrewittBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

FilterPrewittHorizBorder

Filters the image using a horizontal Prewitt filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

NppStatus nppiFilterPrewittHorizBorder_8u_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned horizontal Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus nppiFilterPrewittHorizBorder_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned horizontal Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus nppiFilterPrewittHorizBorder_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus nppiFilterPrewittHorizBorder_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Prewitt filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus nppiFilterPrewittHorizBorder_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed horizontal Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

```
NppStatus nppiFilterPrewittHorizBorder_16s_C3R_Ctx( const Npp16s *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp16s *pDst,
                                                    Npp32s nDstStep, NppiSize oSizeROI,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Three channel 16-bit signed horizontal Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

```
NppStatus nppiFilterPrewittHorizBorder_16s_C4R_Ctx( const Npp16s *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp16s *pDst,
                                                    Npp32s nDstStep, NppiSize oSizeROI,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Four channel 16-bit signed horizontal Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

```
NppStatus nppiFilterPrewittHorizBorder_16s_AC4R_Ctx( const Npp16s *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp16s
                                                    *pDst, Npp32s nDstStep, NppiSize
                                                    oSizeROI, NppiBorderType
                                                    eBorderType, NppStreamContext
                                                    nppStreamCtx)
```

Four channel 16-bit signed horizontal Prewitt filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

```
NppStatus nppiFilterPrewittHorizBorder_32f_C1R_Ctx( const Npp32f *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp32f *pDst,
                                                    Npp32s nDstStep, NppiSize oSizeROI,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Single channel 32-bit floating-point horizontal Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

```
NppStatus nppiFilterPrewittHorizBorder_32f_C3R_Ctx( const Npp32f *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp32f *pDst,
                                                    Npp32s nDstStep, NppiSize oSizeROI,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Three channel 32-bit floating-point horizontal Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

```
NppStatus nppiFilterPrewittHorizBorder_32f_C4R_Ctx(const Npp32f *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp32f *pDst,
                                                    Npp32s nDstStep, NppiSize oSizeROI,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Four channel 32-bit floating-point horizontal Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

```
NppStatus nppiFilterPrewittHorizBorder_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp32f
                                                    *pDst, Npp32s nDstStep, NppiSize
                                                    oSizeROI, NppiBorderType
                                                    eBorderType, NppStreamContext
                                                    nppStreamCtx)
```

Four channel 32-bit floating-point horizontal Prewitt filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

FilterPrewittVertBorder

Filters the image using a vertical Prewitt filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix};$$

```
NppStatus nppiFilterPrewittVertBorder_8u_C1R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,
                                                    NppiSize oSrcSize, NppiPoint
                                                    oSrcOffset, Npp8u *pDst, Npp32s
                                                    nDstStep, NppiSize oSizeROI,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Single channel 8-bit unsigned vertical Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus **nppiFilterPrewittVertBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned vertical Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus **nppiFilterPrewittVertBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned vertical Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus **nppiFilterPrewittVertBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned vertical Prewitt filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus **nppiFilterPrewittVertBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed vertical Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus **nppiFilterPrewittVertBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed vertical Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus nppiFilterPrewittVertBorder_16s_C4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed vertical Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus nppiFilterPrewittVertBorder_16s_AC4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed vertical Prewitt filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus nppiFilterPrewittVertBorder_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point vertical Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus nppiFilterPrewittVertBorder_32f_C3R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point vertical Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

NppStatus nppiFilterPrewittVertBorder_32f_C4R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point vertical Prewitt filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

```
NppStatus nppiFilterPrewittVertBorder_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s
                                                    nSrcStep, NppiSize oSrcSize,
                                                    NppiPoint oSrcOffset, Npp32f *pDst,
                                                    Npp32s nDstStep, NppiSize oSizeROI,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext nppStreamCtx)
```

Four channel 32-bit floating-point vertical Prewitt filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterPrewittBorder functions*.

1.10.5.4 Image Filter Scharr

1.10.5.4.1 FilterScharr

Filters the image using a Scharr filter kernel.

1.10.5.4.1.1 Common parameters for nppiFilterScharr functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

FilterScharrHoriz

Filters the image using a horizontal Scharr filter kernel:

$$\begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$$

NppStatus nppiFilterScharrHoriz_8u16s_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed horizontal Scharr filter.

For common parameter descriptions, see *Common parameters for nppiFilterScharr functions*.

NppStatus nppiFilterScharrHoriz_8s16s_C1R_Ctx(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed horizontal Scharr filter.

For common parameter descriptions, see *Common parameters for nppiFilterScharr functions*.

NppStatus nppiFilterScharrHoriz_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point horizontal Scharr filter.

For common parameter descriptions, see *Common parameters for nppiFilterScharr functions*.

FilterScharrVert

Filters the image using a vertical Scharr filter kernel:

$$\begin{pmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{pmatrix}$$

NppStatus nppiFilterScharrVert_8u16s_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed vertical Scharr filter.

For common parameter descriptions, see *Common parameters for nppiFilterScharr functions*.

NppStatus nppiFilterScharrVert_8s16s_C1R_Ctx(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed vertical Scharr filter.

For common parameter descriptions, see *Common parameters for nppiFilterScharr functions*.

NppStatus nppiFilterScharrVert_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point vertical Scharr filter.

For common parameter descriptions, see *Common parameters for nppiFilterScharr functions*.

1.10.5.5 Image Filter Scharr Border

1.10.5.5.1 FilterScharrBorder

Filters the image using a Scharr filter kernel with border control.

1.10.5.5.1.1 Common parameters for nppiFilterScharrBorder functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param oSrcSize
Source image width and height in pixels relative to pSrc.

param oSrcOffset
The pixel offset that pSrc points to relative to the origin of the source image.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param eBorderType
The border type operation to be applied at source image border boundaries.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

FilterScharrHorizBorder

Filters the image using a horizontal Scharr filter kernel with border control:

$$\begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$$

```
NppStatus nppiFilterScharrHorizBorder_8u16s_C1R_Ctx(const Npp8u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16s
*pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Single channel 8-bit unsigned to 16-bit signed horizontal Scharr filter kernel with border control.

For common parameter descriptions, see *Common parameters for nppiFilterScharrBorder functions:*.

```
NppStatus nppiFilterScharrHorizBorder_8s16s_C1R_Ctx(const Npp8s *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16s
*pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Single channel 8-bit signed to 16-bit signed horizontal Scharr filter kernel with border control.

For common parameter descriptions, see *Common parameters for nppiFilterScharrBorder functions:*.

```
NppStatus nppiFilterScharrHorizBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Single channel 32-bit floating-point horizontal Scharr filter kernel with border control.

For common parameter descriptions, see *Common parameters for nppiFilterScharrBorder functions:*.

FilterScharrVertBorder

Filters the image using a vertical Scharr filter kernel kernel with border control:

$$\begin{pmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{pmatrix}$$

```
NppStatus nppiFilterScharrVertBorder_8u16s_C1R_Ctx(const Npp8u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16s *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Single channel 8-bit unsigned to 16-bit signed vertical Scharr filter kernel with border control.

For common parameter descriptions, see *Common parameters for nppiFilterScharrBorder functions:*.

```
NppStatus nppiFilterScharrVertBorder_8s16s_C1R_Ctx(const Npp8s *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16s *pDst,
Npp32s nDstStep, NppiSize oSizeROI,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Single channel 8-bit signed to 16-bit signed vertical Scharr filter kernel with border control.

For common parameter descriptions, see *Common parameters for nppiFilterScharrBorder functions*:

NppStatus **nppiFilterScharrVertBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point vertical Scharr filter kernel with border control.

For common parameter descriptions, see *Common parameters for nppiFilterScharrBorder functions*:

1.10.5.6 Image Filter Sobel

1.10.5.6.1 FilterSobel

Filters the image using a Sobel filter kernel.

1.10.5.6.1.1 Common parameters for nppiFilterSobel functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

Enumeration value specifying the mask size.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

FilterSobelHoriz

Filters the image using a horizontal Sobel filter kernel:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{pmatrix}$$

NppStatus nppiFilterSobelHoriz_8u_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus nppiFilterSobelHoriz_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus nppiFilterSobelHoriz_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus nppiFilterSobelHoriz_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed horizontal Sobel filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus nppiFilterSobelHoriz_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus nppiFilterSobelHoriz_16s_C3R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus **nppiFilterSobelHoriz_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus **nppiFilterSobelHoriz_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Sobel filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus **nppiFilterSobelHoriz_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus **nppiFilterSobelHoriz_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus **nppiFilterSobelHoriz_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus **nppiFilterSobelHoriz_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point horizontal Sobel filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus **nppiFilterSobelHoriz_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus **nppiFilterSobelHoriz_8s16s_C1R_Ctx**(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*:

NppStatus nppiFilterSobelHorizMask_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

FilterSobelVert

Filters the image using a vertical Sobel filter kernel:

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{pmatrix}$$

NppStatus nppiFilterSobelVert_8u_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus nppiFilterSobelVert_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus nppiFilterSobelVert_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus nppiFilterSobelVert_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed vertical Sobel filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus nppiFilterSobelVert_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus **nppiFilterSobelVert_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus **nppiFilterSobelVert_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus **nppiFilterSobelVert_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned vertical Sobel filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus **nppiFilterSobelVert_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus **nppiFilterSobelVert_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus **nppiFilterSobelVert_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus **nppiFilterSobelVert_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point vertical Sobel filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus **nppiFilterSobelVert_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus nppiFilterSobelVert_8s16s_C1R_Ctx(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus nppiFilterSobelVertMask_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point vertical Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

FilterSobelHorizSecond

Filters the image using a second derivative, horizontal Sobel filter kernel:

$$\begin{pmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

NppStatus nppiFilterSobelHorizSecond_8u16s_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed second derivative, horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus nppiFilterSobelHorizSecond_8s16s_C1R_Ctx(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed second derivative, horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus nppiFilterSobelHorizSecond_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point second derivative, horizontal Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

FilterSobelVertSecond

Filters the image using a second derivative, vertical Sobel filter kernel:

$$\begin{pmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 & 0 & 1 \\ 4 & 0 & -8 & 0 & 4 \\ 6 & 0 & -12 & 0 & 6 \\ 4 & 0 & -8 & 0 & 4 \\ 1 & 0 & -2 & 0 & 1 \end{pmatrix}$$

NppStatus nppiFilterSobelVertSecond_8u16s_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed second derivative, vertical Sobel filter.

For common parameter descriptions, see [Common parameters for nppiFilterSobel functions](#).

NppStatus nppiFilterSobelVertSecond_8s16s_C1R_Ctx(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed second derivative, vertical Sobel filter.

For common parameter descriptions, see [Common parameters for nppiFilterSobel functions](#).

NppStatus nppiFilterSobelVertSecond_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point second derivative, vertical Sobel filter.

For common parameter descriptions, see [Common parameters for nppiFilterSobel functions](#).

FilterSobelCross

Filters the image using a second cross derivative Sobel filter kernel:

$$\begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -4 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 0 & -4 & -2 \\ 1 & 2 & 0 & -2 & -1 \end{pmatrix}$$

NppStatus **nppiFilterSobelCross_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed second cross derivative Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus **nppiFilterSobelCross_8s16s_C1R_Ctx**(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed second cross derivative Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

NppStatus **nppiFilterSobelCross_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point second cross derivative Sobel filter.

For common parameter descriptions, see *Common parameters for nppiFilterSobel functions*.

1.10.5.7 Image Filter Sobel Border

1.10.5.7.1 FilterSobelBorder

Filters the image using a Sobel filter kernel with border control.

1.10.5.7.1.1 Common parameters for nppiFilterSobelBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

Enumeration value specifying the mask size.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

FilterSobelHorizBorder

Filters the image using a horizontal Sobel filter kernel with border control:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{pmatrix}$$

NppStatus **nppiFilterSobelHorizBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed horizontal Sobel filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Sobel filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point horizontal Sobel filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelHorizBorder_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.


```
NppStatus nppiFilterSobelHorizBorder_8s16s_C1R_Ctx(const Npp8s *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16s *pDst,
Npp32s nDstStep, NppiSize
oSizeROI, NppiMaskSize eMaskSize,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Single channel 8-bit signed to 16-bit signed horizontal Sobel filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterSobelBorder functions](#).

```
NppStatus nppiFilterSobelHorizMaskBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f
*pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiMaskSize
eMaskSize, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Single channel 32-bit floating-point horizontal Sobel filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterSobelBorder functions](#).

FilterSobelVertBorder

Filters the image using a vertical Sobel filter kernel with border control:

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{pmatrix}$$

```
NppStatus nppiFilterSobelVertBorder_8u_C1R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp8u *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Single channel 8-bit unsigned vertical Sobel filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterSobelBorder functions](#).

```
NppStatus nppiFilterSobelVertBorder_8u_C3R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp8u *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Three channel 8-bit unsigned vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions:*.

```
NppStatus nppiFilterSobelVertBorder_8u_C4R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp8u *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, NppiBorderType eBorderType,  
                                                NppStreamContext nppStreamCtx)
```

Four channel 8-bit unsigned vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions:*.

```
NppStatus nppiFilterSobelVertBorder_8u_AC4R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp8u *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, NppiBorderType eBorderType,  
                                                NppStreamContext nppStreamCtx)
```

Four channel 16-bit signed vertical Sobel filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions:*.

```
NppStatus nppiFilterSobelVertBorder_16s_C1R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp16s *pDst, Npp32s nDstStep,  
                                                NppiSize oSizeROI, NppiBorderType  
                                                eBorderType, NppStreamContext  
                                                nppStreamCtx)
```

Single channel 16-bit signed vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions:*.

```
NppStatus nppiFilterSobelVertBorder_16s_C3R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp16s *pDst, Npp32s nDstStep,  
                                                NppiSize oSizeROI, NppiBorderType  
                                                eBorderType, NppStreamContext  
                                                nppStreamCtx)
```

Three channel 16-bit signed vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions:*.

```
NppStatus nppiFilterSobelVertBorder_16s_C4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp16s *pDst, Npp32s nDstStep,  
                                                NppiSize oSizeROI, NppiBorderType  
                                                eBorderType, NppStreamContext  
                                                nppStreamCtx)
```

Four channel 16-bit signed vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions:*.

NppStatus **nppiFilterSobelVertBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned vertical Sobel filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelVertBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelVertBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelVertBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelVertBorder_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point vertical Sobel filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelVertBorder_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelVertBorder_8s16s_C1R_Ctx**(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

NppStatus **nppiFilterSobelVertMaskBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point vertical Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

FilterSobelHorizSecondBorder

Filters the image using a second derivative, horizontal Sobel filter kernel with border control:

$$\begin{pmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

```
NppStatus nppiFilterSobelHorizSecondBorder_8u16s_C1R_Ctx( const Npp8u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset,
Npp16s *pDst, Npp32s
nDstStep, NppiSize oSizeROI,
NppiMaskSize eMaskSize,
NppiBorderType
eBorderType,
NppStreamContext
nppStreamCtx)
```

Single channel 8-bit unsigned to 16-bit signed second derivative, horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

```
NppStatus nppiFilterSobelHorizSecondBorder_8s16s_C1R_Ctx( const Npp8s *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset,
Npp16s *pDst, Npp32s
nDstStep, NppiSize oSizeROI,
NppiMaskSize eMaskSize,
NppiBorderType
eBorderType,
NppStreamContext
nppStreamCtx)
```

Single channel 8-bit signed to 16-bit signed second derivative, horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

```
NppStatus nppiFilterSobelHorizSecondBorder_32f_C1R_Ctx( const Npp32f *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f
*pDst, Npp32s nDstStep,
NppiSize oSizeROI,
NppiMaskSize eMaskSize,
NppiBorderType eBorderType,
NppStreamContext
nppStreamCtx)
```

Single channel 32-bit floating-point second derivative, horizontal Sobel filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSobelBorder functions*.

FilterSobelVertSecondBorder

Filters the image using a second derivative, vertical Sobel filter kernel with border control:

$$\begin{pmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 & 0 & 1 \\ 4 & 0 & -8 & 0 & 4 \\ 6 & 0 & -12 & 0 & 6 \\ 4 & 0 & -8 & 0 & 4 \\ 1 & 0 & -2 & 0 & 1 \end{pmatrix}$$

NppStatus **nppiFilterSobelVertSecondBorder_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed second derivative, vertical Sobel filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterSobelBorder functions](#).

NppStatus **nppiFilterSobelVertSecondBorder_8s16s_C1R_Ctx**(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed second derivative, vertical Sobel filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterSobelBorder functions](#).

NppStatus **nppiFilterSobelVertSecondBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point second derivative, vertical Sobel filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterSobelBorder functions](#).

FilterSobelCrossBorder

Filters the image using a second cross derivative Sobel filter kernel with border control:

$$\begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -4 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 0 & -4 & -2 \\ 1 & 2 & 0 & -2 & -1 \end{pmatrix}$$

NppStatus nppiFilterSobelCrossBorder_8u16s_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed second cross derivative Sobel filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterSobelBorder functions](#).

NppStatus nppiFilterSobelCrossBorder_8s16s_C1R_Ctx(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed second cross derivative Sobel filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterSobelBorder functions](#).

NppStatus nppiFilterSobelCrossBorder_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point second cross derivative Sobel filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterSobelBorder functions](#).

1.10.5.8 Image Filter Roberts

1.10.5.8.1 FilterRoberts

Filters the image using a Roberts filter kernel.

1.10.5.8.1.1 Common parameters for nppiFilterRoberts functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

FilterRobertsDown

Filters the image using a horizontal Roberts filter kernel:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

NppStatus **nppiFilterRobertsDown_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned horizontal Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions:*

NppStatus **nppiFilterRobertsDown_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned horizontal Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions:*

NppStatus nppiFilterRobertsDown_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus nppiFilterRobertsDown_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Roberts filter, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus nppiFilterRobertsDown_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed horizontal Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus nppiFilterRobertsDown_16s_C3R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed horizontal Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus nppiFilterRobertsDown_16s_C4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed horizontal Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus nppiFilterRobertsDown_16s_AC4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed horizontal Roberts filter, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus nppiFilterRobertsDown_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point horizontal Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus nppiFilterRobertsDown_32f_C3R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point horizontal Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsDown_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point horizontal Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsDown_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point horizontal Roberts filter, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

FilterRobertsUp

Filters the image using a vertical Roberts filter kernel:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

NppStatus **nppiFilterRobertsUp_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned vertical Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned vertical Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned vertical Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned vertical Roberts filter, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed vertical Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed vertical Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed vertical Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed vertical Roberts filter, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point vertical Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point vertical Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point vertical Roberts filter.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

NppStatus **nppiFilterRobertsUp_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point vertical Roberts filter, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRoberts functions*.

1.10.5.9 Image Filter Roberts Border

1.10.5.9.1 FilterRobertsBorder

Filters the image using a Roberts filter kernel with border control.

1.10.5.9.1.1 Common parameters for nppiFilterRobertsBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

FilterRobertsDownBorder

Filters the image using a horizontal Roberts filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

NppStatus nppiFilterRobertsDownBorder_8u_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned horizontal Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus nppiFilterRobertsDownBorder_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned horizontal Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus nppiFilterRobertsDownBorder_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus nppiFilterRobertsDownBorder_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned horizontal Roberts filter with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus nppiFilterRobertsDownBorder_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed horizontal Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus **nppiFilterRobertsDownBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed horizontal Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus **nppiFilterRobertsDownBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed horizontal Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus **nppiFilterRobertsDownBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed horizontal Roberts filter with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus **nppiFilterRobertsDownBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point horizontal Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus **nppiFilterRobertsDownBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point horizontal Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus nppiFilterRobertsDownBorder_32f_C4R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point horizontal Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus nppiFilterRobertsDownBorder_32f_AC4R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point horizontal Roberts filter with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

FilterRobertsUpBorder

Filters the image using a vertical Roberts filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

NppStatus nppiFilterRobertsUpBorder_8u_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned vertical Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus nppiFilterRobertsUpBorder_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned vertical Roberts filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterRobertsBorder functions:](#).

```
NppStatus nppiFilterRobertsUpBorder_8u_C4R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp8u *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, NppiBorderType eBorderType,  
                                                NppStreamContext nppStreamCtx)
```

Four channel 8-bit unsigned vertical Roberts filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterRobertsBorder functions:](#).

```
NppStatus nppiFilterRobertsUpBorder_8u_AC4R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp8u *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, NppiBorderType eBorderType,  
                                                NppStreamContext nppStreamCtx)
```

Four channel 8-bit unsigned vertical Roberts filter with border control, ignoring alpha-channel.

For common parameter descriptions, see [Common parameters for nppiFilterRobertsBorder functions:](#).

```
NppStatus nppiFilterRobertsUpBorder_16s_C1R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp16s *pDst, Npp32s nDstStep,  
                                                NppiSize oSizeROI, NppiBorderType  
                                                eBorderType, NppStreamContext  
                                                nppStreamCtx)
```

Single channel 16-bit signed vertical Roberts filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterRobertsBorder functions:](#).

```
NppStatus nppiFilterRobertsUpBorder_16s_C3R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp16s *pDst, Npp32s nDstStep,  
                                                NppiSize oSizeROI, NppiBorderType  
                                                eBorderType, NppStreamContext  
                                                nppStreamCtx)
```

Three channel 16-bit signed vertical Roberts filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterRobertsBorder functions:](#).

```
NppStatus nppiFilterRobertsUpBorder_16s_C4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp16s *pDst, Npp32s nDstStep,  
                                                NppiSize oSizeROI, NppiBorderType  
                                                eBorderType, NppStreamContext  
                                                nppStreamCtx)
```

Four channel 16-bit signed vertical Roberts filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterRobertsBorder functions:](#).

NppStatus **nppiFilterRobertsUpBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed vertical Roberts filter with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus **nppiFilterRobertsUpBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point vertical Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus **nppiFilterRobertsUpBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point vertical Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus **nppiFilterRobertsUpBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point vertical Roberts filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

NppStatus **nppiFilterRobertsUpBorder_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point vertical Roberts filter with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiFilterRobertsBorder functions*.

1.10.5.10 Image Filter Laplace

1.10.5.10.1 FilterLaplace

Filters the image using a Laplacian filter kernel.

1.10.5.10.1.1 Common parameters for nppiFilterLaplace functions:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -3 & -4 & -3 & -1 \\ -3 & 0 & 6 & 0 & -3 \\ -4 & 6 & 20 & 6 & -4 \\ -3 & 0 & 6 & 0 & -3 \\ -1 & -3 & -4 & -3 & -1 \end{pmatrix}$$

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

Enumeration value specifying the mask size.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterLaplace_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions:*.

NppStatus **nppiFilterLaplace_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*:

NppStatus **nppiFilterLaplace_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*:

NppStatus **nppiFilterLaplace_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Laplace filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*:

NppStatus **nppiFilterLaplace_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*:

NppStatus **nppiFilterLaplace_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*:

NppStatus **nppiFilterLaplace_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*:

NppStatus **nppiFilterLaplace_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Laplace filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*:

NppStatus **nppiFilterLaplace_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*:

NppStatus **nppiFilterLaplace_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*.

NppStatus **nppiFilterLaplace_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*.

NppStatus **nppiFilterLaplace_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point Laplace filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*.

NppStatus **nppiFilterLaplace_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*.

NppStatus **nppiFilterLaplace_8s16s_C1R_Ctx**(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed Laplace filter.

For common parameter descriptions, see *Common parameters for nppiFilterLaplace functions*.

1.10.5.11 Image Filter Laplace Border

1.10.5.11.1 FilterLaplaceBorder

Filters the image using a Laplacian filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.5.11.1.1 Common parameters for nppiFilterLaplaceBorder functions:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -3 & -4 & -3 & -1 \\ -3 & 0 & 6 & 0 & -3 \\ -4 & 6 & 20 & 6 & -4 \\ -3 & 0 & 6 & 0 & -3 \\ -1 & -3 & -4 & -3 & -1 \end{pmatrix}$$

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

Enumeration value specifying the mask size.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterLaplaceBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Laplace filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Laplace filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point Laplace filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned to 16-bit signed Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

NppStatus **nppiFilterLaplaceBorder_8s16s_C1R_Ctx**(const *Npp8s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit signed to 16-bit signed Laplace filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterLaplaceBorder functions*.

1.10.5.12 Image Filter Gauss

1.10.5.12.1 FilterGauss

Filters the image using a Gaussian filter kernel. Use FilterGaussAdvanced if you want to supply your own filter coefficients.

Note that all FilterGauss functions currently support mask sizes up to 15x15. Filter kernels for these functions are calculated using a sigma value of $0.4F + (\text{mask width} / 2) * 0.6F$.

1.10.5.12.1.1 Common parameters for nppiFilterGauss functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param eMaskSize
 Enumeration value specifying the mask size.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterGauss_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*.

NppStatus **nppiFilterGauss_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*.

NppStatus **nppiFilterGauss_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*.

NppStatus **nppiFilterGauss_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Gauss filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*.

NppStatus **nppiFilterGauss_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*..

NppStatus **nppiFilterGauss_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*..

NppStatus **nppiFilterGauss_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*..

NppStatus **nppiFilterGauss_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned Gauss filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*..

NppStatus **nppiFilterGauss_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*..

NppStatus **nppiFilterGauss_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*..

NppStatus **nppiFilterGauss_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*..

NppStatus **nppiFilterGauss_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Gauss filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*..

NppStatus nppiFilterGauss_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*.

NppStatus nppiFilterGauss_32f_C3R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*.

NppStatus nppiFilterGauss_32f_C4R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*.

NppStatus nppiFilterGauss_32f_AC4R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point Gauss filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGauss functions*.

1.10.5.13 Image Filter Gauss Advanced

1.10.5.13.1 FilterGaussAdvanced

Filters the image using a separable Gaussian filter kernel with user supplied floating point coefficients:

1.10.5.13.1.1 Common parameters for nppiFilterGaussAdvanced functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nFilterTaps

The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

param pKernel

Pointer to an array of nFilterTaps kernel coefficients which sum to 1.OF.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterGaussAdvanced_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

NppStatus **nppiFilterGaussAdvanced_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

NppStatus **nppiFilterGaussAdvanced_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

NppStatus **nppiFilterGaussAdvanced_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Gauss filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

NppStatus **nppiFilterGaussAdvanced_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

NppStatus nppiFilterGaussAdvanced_16u_C3R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions*.

NppStatus nppiFilterGaussAdvanced_16u_C4R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions*.

NppStatus nppiFilterGaussAdvanced_16u_AC4R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned Gauss filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions*.

NppStatus nppiFilterGaussAdvanced_16s_C1R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions*.

NppStatus nppiFilterGaussAdvanced_16s_C3R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions*.

NppStatus nppiFilterGaussAdvanced_16s_C4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

```
NppStatus nppiFilterGaussAdvanced_16s_AC4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                                Npp16s *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, const int nFilterTaps, const  
                                                Npp32f *pKernel, NppStreamContext  
                                                nppStreamCtx)
```

Four channel 16-bit signed Gauss filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

```
NppStatus nppiFilterGaussAdvanced_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, const int nFilterTaps, const  
                                                Npp32f *pKernel, NppStreamContext  
                                                nppStreamCtx)
```

Single channel 32-bit floating-point Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

```
NppStatus nppiFilterGaussAdvanced_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, const int nFilterTaps, const  
                                                Npp32f *pKernel, NppStreamContext  
                                                nppStreamCtx)
```

Three channel 32-bit floating-point Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

```
NppStatus nppiFilterGaussAdvanced_32f_C4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, const int nFilterTaps, const  
                                                Npp32f *pKernel, NppStreamContext  
                                                nppStreamCtx)
```

Four channel 32-bit floating-point Gauss filter.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

```
NppStatus nppiFilterGaussAdvanced_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, const int nFilterTaps, const  
                                                Npp32f *pKernel, NppStreamContext  
                                                nppStreamCtx)
```

Four channel 32-bit floating-point Gauss filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvanced functions:*.

1.10.5.14 Image Filter Gauss Border

1.10.5.14.1 FilterGaussBorder

Filters the image using a Gaussian filter kernel with border control. Use FilterGaussAdvancedBorder if you want to supply your own filter coefficients.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

Note that all FilterGaussBorder functions currently support mask sizes up to 15x15. Filter kernels for these functions are calculated using a sigma value of $0.4F + (\text{mask width} / 2) * 0.6F$.

1.10.5.14.1.1 Common parameters for nppiFilterGaussBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Full source image width and height in pixels.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

Enumeration value specifying the mask size.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterGaussBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Gauss filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned Gauss filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Gauss filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

NppStatus **nppiFilterGaussBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*.

```
NppStatus nppiFilterGaussBorder_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep, NppiSize
oSizeROI, NppiMaskSize eMaskSize,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 32-bit floating-point Gauss filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussBorder functions*:

1.10.5.15 Image Filter Advanced Gauss Border

1.10.5.15.1 FilterGaussAdvancedBorder

Filters the image using a separable Gaussian filter kernel with user supplied floating point coefficients with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Note that the performance of these functions can drop significantly for filter kernels with a very large number of taps.

Currently only the NPP_BORDER_REPLICATE and NPP_BORDER_MIRROR border type operations are supported.

1.10.5.15.1.1 Common parameters for nppiFilterGaussAdvancedBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Full source image width and height in pixels.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nFilterTaps

The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

param pKernel

Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterGaussAdvancedBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

```
NppStatus nppiFilterGaussAdvancedBorder_8u_AC4R_Ctx(const Npp8u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp8u *pDst,
Npp32s nDstStep, NppiSize
oSizeROI, const int nFilterTaps,
const Npp32f *pKernel,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 8-bit unsigned Gauss filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

```
NppStatus nppiFilterGaussAdvancedBorder_16u_C1R_Ctx(const Npp16u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16u
*pDst, Npp32s nDstStep, NppiSize
oSizeROI, const int nFilterTaps,
const Npp32f *pKernel,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Single channel 16-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

```
NppStatus nppiFilterGaussAdvancedBorder_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16u
*pDst, Npp32s nDstStep, NppiSize
oSizeROI, const int nFilterTaps,
const Npp32f *pKernel,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Three channel 16-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

```
NppStatus nppiFilterGaussAdvancedBorder_16u_C4R_Ctx(const Npp16u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16u
*pDst, Npp32s nDstStep, NppiSize
oSizeROI, const int nFilterTaps,
const Npp32f *pKernel,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned Gauss filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Gauss filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*.

NppStatus **nppiFilterGaussAdvancedBorder_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point Gauss filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterGaussAdvancedBorder functions*:

1.10.5.16 Image Filter Gauss Pyramid Layer Down Border

1.10.5.16.1 FilterGaussPyramidLayerDownBorder

Filters the image using a separable Gaussian filter kernel with user supplied floating point coefficients with downsampling and border control.

If the downsampling rate is equivalent to an integer value then unnecessary source pixels are just skipped. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_MIRROR and NPP_BORDER_REPLICATE border type operations are supported.

1.10.5.16.1.1 Common parameters for nppiFilterGaussPyramidLayerDownBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nRate

The downsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.

param nFilterTaps

The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

param pKernel

Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiGetFilterGaussPyramidLayerDownBorderDstROI** (int nSrcROIWidth, int nSrcROIHeight, *NppiSize* *pDstSizeROI, *Npp32f* nRate)

Calculate destination image SizeROI width and height from source image ROI width and height and downsampling rate.

It is highly recommended that this function be use to determine the destination image ROI for consistent results.

Parameters

- ▶ **nSrcROIWidth** – The desired source image ROI width, must be $\leq$ oSrc-Size.width.
- ▶ **nSrcROIHeight** – The desired source image ROI height, must be $\leq$ oSrc-Size.height.
- ▶ **pDstSizeROI** – Host memory pointer to the destination image roi_specification.
- ▶ **nRate** – The downsampling or upsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be $> 1.0F$ and $\leq 10.0F$.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiFilterGaussPyramidLayerDownBorder_8u_C1R_Ctx** (const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRate, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned Gauss filter with downsampling and border control.

For common parameter descriptions, see [Common parameters for nppiFilterGaussPyramidLayerDownBorder functions](#).

```
NppStatus nppiFilterGaussPyramidLayerDownBorder_8u_C3R_Ctx( const Npp8u *pSrc,  
                                                             Npp32s nSrcStep,  
                                                             NppiSize oSrcSize,  
                                                             NppiPoint oSrcOffset,  
                                                             Npp8u *pDst, Npp32s  
                                                             nDstStep, NppiSize  
                                                             oSizeROI, Npp32f nRate,  
                                                             const int nFilterTaps,  
                                                             const Npp32f *pKernel,  
                                                             NppiBorderType  
                                                             eBorderType,  
                                                             NppStreamContext  
                                                             nppStreamCtx )
```

Three channel 8-bit unsigned Gauss filter with downsampling and border control.

For common parameter descriptions, see [Common parameters for nppiFilterGaussPyramidLayerDownBorder functions](#).

```
NppStatus nppiFilterGaussPyramidLayerDownBorder_16u_C1R_Ctx( const Npp16u *pSrc,  
                                                             Npp32s nSrcStep,  
                                                             NppiSize oSrcSize,  
                                                             NppiPoint oSrcOffset,  
                                                             Npp16u *pDst, Npp32s  
                                                             nDstStep, NppiSize  
                                                             oSizeROI, Npp32f nRate,  
                                                             const int nFilterTaps,  
                                                             const Npp32f *pKernel,  
                                                             NppiBorderType  
                                                             eBorderType,  
                                                             NppStreamContext  
                                                             nppStreamCtx )
```

Single channel 16-bit unsigned Gauss filter with downsampling and border control.

For common parameter descriptions, see [Common parameters for nppiFilterGaussPyramidLayerDownBorder functions](#).

```
NppStatus nppiFilterGaussPyramidLayerDownBorder_16u_C3R_Ctx( const Npp16u *pSrc,  
                                                             Npp32s nSrcStep,  
                                                             NppiSize oSrcSize,  
                                                             NppiPoint oSrcOffset,  
                                                             Npp16u *pDst, Npp32s  
                                                             nDstStep, NppiSize  
                                                             oSizeROI, Npp32f nRate,  
                                                             const int nFilterTaps,  
                                                             const Npp32f *pKernel,  
                                                             NppiBorderType  
                                                             eBorderType,  
                                                             NppStreamContext  
                                                             nppStreamCtx )
```

Three channel 16-bit unsigned Gauss filter with downsampling and border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussPyramidLayerDownBorder functions*.

```
NppStatus nppiFilterGaussPyramidLayerDownBorder_32f_C1R_Ctx(const Npp32f *pSrc,
                                                             Npp32s nSrcStep,
                                                             NppiSize oSrcSize,
                                                             NppiPoint oSrcOffset,
                                                             Npp32f *pDst, Npp32s
                                                             nDstStep, NppiSize
                                                             oSizeROI, Npp32f nRate,
                                                             const int nFilterTaps,
                                                             const Npp32f *pKernel,
                                                             NppiBorderType
                                                             eBorderType,
                                                             NppStreamContext
                                                             nppStreamCtx)
```

Single channel 32-bit floating-point Gauss filter downsampling and with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussPyramidLayerDownBorder functions*.

```
NppStatus nppiFilterGaussPyramidLayerDownBorder_32f_C3R_Ctx(const Npp32f *pSrc,
                                                             Npp32s nSrcStep,
                                                             NppiSize oSrcSize,
                                                             NppiPoint oSrcOffset,
                                                             Npp32f *pDst, Npp32s
                                                             nDstStep, NppiSize
                                                             oSizeROI, Npp32f nRate,
                                                             const int nFilterTaps,
                                                             const Npp32f *pKernel,
                                                             NppiBorderType
                                                             eBorderType,
                                                             NppStreamContext
                                                             nppStreamCtx)
```

Three channel 32-bit floating-point Gauss filter with downsampling and border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussPyramidLayerDownBorder functions*.

1.10.5.17 Image Filter Gauss Pyramid Layer Up Border

1.10.5.17.1 FilterGaussPyramidLayerUpBorder

Filters the image using a separable Gaussian filter kernel with user supplied floating point coefficients with upsampling and border control.

If the upsampling rate is equivalent to an integer value then unnecessary source pixels are just skipped. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_MIRROR and NPP_BORDER_REPLICATE border type operations are supported.

1.10.5.17.1.1 Common parameters for nppiFilterGaussPyramidLayerUpBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nRate

The upsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be > 1.0F and <= 10.0F.

param nFilterTaps

The number of filter taps where $nFilterTaps = 2 * ((int)((float)ceil(radius) + 0.5F)) + 1$.

param pKernel

Pointer to an array of nFilterTaps kernel coefficients which sum to 1.0F.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiGetFilterGaussPyramidLayerUpBorderDstROI** (int nSrcROIWidth, int nSrcROIHeight, *NppiSize* *pDstSizeROIMin, *NppiSize* *pDstSizeROIMax, *Npp32f* nRate)

Calculate destination image minimum and maximum SizeROI width and height from source image ROI width and height and upsampling rate.

It is highly recommended that this function be use to determine the best destination image ROI for consistent results.

Parameters

- ▶ **nSrcROIWidth** – The desired source image ROI width, must be $\leq$ oSrc-Size.width.
- ▶ **nSrcROIHeight** – The desired source image ROI height, must be $\leq$ oSrc-Size.height.
- ▶ **pDstSizeROImin** – Host memory pointer to the minimum recommended destination image roi_specification.
- ▶ **pDstSizeROImax** – Host memory pointer to the maximum recommended destination image roi_specification.
- ▶ **nRate** – The upsampling rate to be used. For integer equivalent rates unnecessary source pixels are just skipped. For non-integer rates the source image is bilinear interpolated. nRate must be $> 1.0F$ and $\leq 10.0F$.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiFilterGaussPyramidLayerUpBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRate, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned Gauss filter with upsampling and border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussPyramid-LayerUpBorder functions*.

NppStatus **nppiFilterGaussPyramidLayerUpBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRate, const int nFilterTaps, const *Npp32f* *pKernel, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned Gauss filter with upsampling and border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussPyramid-LayerUpBorder functions*.

```
NppStatus nppiFilterGaussPyramidLayerUpBorder_16u_C1R_Ctx(const Npp16u *pSrc,  
                                                           Npp32s nSrcStep, NppiSize  
                                                           oSrcSize, NppiPoint  
                                                           oSrcOffset, Npp16u *pDst,  
                                                           Npp32s nDstStep, NppiSize  
                                                           oSizeROI, Npp32f nRate,  
                                                           const int nFilterTaps, const  
                                                           Npp32f *pKernel,  
                                                           NppiBorderType  
                                                           eBorderType,  
                                                           NppStreamContext  
                                                           nppStreamCtx)
```

Single channel 16-bit unsigned Gauss filter with upsampling and border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussPyramid-LayerUpBorder functions*.

```
NppStatus nppiFilterGaussPyramidLayerUpBorder_16u_C3R_Ctx(const Npp16u *pSrc,  
                                                           Npp32s nSrcStep, NppiSize  
                                                           oSrcSize, NppiPoint  
                                                           oSrcOffset, Npp16u *pDst,  
                                                           Npp32s nDstStep, NppiSize  
                                                           oSizeROI, Npp32f nRate,  
                                                           const int nFilterTaps, const  
                                                           Npp32f *pKernel,  
                                                           NppiBorderType  
                                                           eBorderType,  
                                                           NppStreamContext  
                                                           nppStreamCtx)
```

Three channel 16-bit unsigned Gauss filter with upsampling and border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussPyramid-LayerUpBorder functions*.

```
NppStatus nppiFilterGaussPyramidLayerUpBorder_32f_C1R_Ctx(const Npp32f *pSrc,  
                                                           Npp32s nSrcStep, NppiSize  
                                                           oSrcSize, NppiPoint  
                                                           oSrcOffset, Npp32f *pDst,  
                                                           Npp32s nDstStep, NppiSize  
                                                           oSizeROI, Npp32f nRate,  
                                                           const int nFilterTaps, const  
                                                           Npp32f *pKernel,  
                                                           NppiBorderType  
                                                           eBorderType,  
                                                           NppStreamContext  
                                                           nppStreamCtx)
```

Single channel 32-bit floating-point Gauss filter upsampling and with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussPyramid-LayerUpBorder functions*.

```

NppStatus nppiFilterGaussPyramidLayerUpBorder_32f_C3R_Ctx( const Npp32f *pSrc,
                                                            Npp32s nSrcStep, NppiSize
                                                            oSrcSize, NppiPoint
                                                            oSrcOffset, Npp32f *pDst,
                                                            Npp32s nDstStep, NppiSize
                                                            oSizeROI, Npp32f nRate,
                                                            const int nFilterTaps, const
                                                            Npp32f *pKernel,
                                                            NppiBorderType
                                                            eBorderType,
                                                            NppStreamContext
                                                            nppStreamCtx )

```

Three channel 32-bit floating-point Gauss filter with upsampling and border control.

For common parameter descriptions, see *Common parameters for nppiFilterGaussPyramid-LayerUpBorder functions*.

1.10.5.18 Image Filter Bilateral Gauss Border

1.10.5.18.1 FilterBilateralGaussBorder

Filters the image using a bilateral Gaussian filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

For this filter the anchor point is always the central element of the kernel. Coefficients of the bilateral filter kernel depend on their position in the kernel and on the value of some source image pixels overlayed by the filter kernel. Only source image pixels with both coordinates divisible by nDistance-BetweenSrcPixels are used in calculations.

The value of an output pixel d is

$$d = \frac{\sum_{h=-nRadius}^{nRadius} \sum_{w=-nRadius}^{nRadius} W1(h, w) \cdot W2(h, w) \cdot S(h, w)}{\sum_{h=-nRadius}^{nRadius} \sum_{w=-nRadius}^{nRadius} W1(h, w) \cdot W2(h, w)}$$

where h and w are the corresponding kernel width and height indexes, $S(h, w)$ is the value of the source image pixel overlayed by filter kernel position (h, w) , $W1(h, w)$ is $\text{func}(\text{nValSquareSigma}, (S(h, w) - S(0, 0)))$ where $S(0, 0)$ is the value of the source image pixel at the center of the kernel, $W2(h, w)$ is $\text{func}(\text{nPosSquareSigma}, \text{sqrt}(h^2 + w^2))$, and func is the following formula

$$\text{func}(S, I) = \exp\left(-\frac{I^2}{2.0F \cdot S^2}\right)$$

Supported border types:

- ▶ NPP_BORDER_REPLICATE — edge pixel is repeated (clamp)
- ▶ NPP_BORDER_MIRROR (4) — IPP mirrored / OpenCV BORDER_REFLECT_101: gfedcb|abcdefgh|gfedcba (edge NOT repeated)
- ▶ NPP_BORDER_MIRROR_REFL (5) — IPP mirrored with replication / OpenCV BORDER_REFLECT: fedcba|abcdefgh|hgfedcba (edge IS repeated)

1.10.5.18.1.1 Common parameters for nppiFilterBilateralGaussBorder functions:**param pSrc***Source-Image Pointer.***param nSrcStep***Source-Image Line Step.***param oSrcSize**

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst*Destination-Image Pointer.***param nDstStep***Destination-Image Line Step.***param oSizeROI***Region-Of-Interest (ROI).***param nRadius**

The radius of the round filter kernel to be used. A radius of 1 indicates a filter kernel size of 3 by 3, 2 indicates 5 by 5, etc. Radius values from 1 to 32 are supported.

param nStepBetweenSrcPixels

The step size between adjacent source image pixels processed by the filter kernel, most commonly 1.

param nValSquareSigma

The square of the sigma for the relative intensity distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

param nPosSquareSigma

The square of the sigma for the relative geometric distance between a source image pixel in the filter kernel and the source image pixel at the center of the filter kernel.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes*

Functions

NppStatus **nppiFilterBilateralGaussBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const *Npp32f* nValSquareSigma, const *Npp32f* nPosSquareSigma, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned bilateral Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBilateralGaussBorder functions*.

NppStatus **nppiFilterBilateralGaussBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const *Npp32f* nValSquareSigma, const *Npp32f* nPosSquareSigma, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned bilateral Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBilateralGaussBorder functions*.

NppStatus **nppiFilterBilateralGaussBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const int nRadius, const int nStepBetweenSrcPixels, const *Npp32f* nValSquareSigma, const *Npp32f* nPosSquareSigma, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned bilateral Gauss filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterBilateralGaussBorder functions*.

```
NppStatus nppiFilterBilateralGaussBorder_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16u
*pDst, Npp32s nDstStep, NppiSize
oSizeROI, const int nRadius, const
int nStepBetweenSrcPixels, const
Npp32f nValSquareSigma, const
Npp32f nPosSquareSigma,
NppiBorderType eBorderType,
NppStreamContext
nppStreamCtx)
```

Three channel 16-bit unsigned bilateral Gauss filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterBilateralGaussBorder functions](#).

```
NppStatus nppiFilterBilateralGaussBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f
*pDst, Npp32s nDstStep, NppiSize
oSizeROI, const int nRadius, const
int nStepBetweenSrcPixels, const
Npp32f nValSquareSigma, const
Npp32f nPosSquareSigma,
NppiBorderType eBorderType,
NppStreamContext
nppStreamCtx)
```

One channel 32-bit floating-point bilateral Gauss filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterBilateralGaussBorder functions](#).

```
NppStatus nppiFilterBilateralGaussBorder_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f
*pDst, Npp32s nDstStep, NppiSize
oSizeROI, const int nRadius, const
int nStepBetweenSrcPixels, const
Npp32f nValSquareSigma, const
Npp32f nPosSquareSigma,
NppiBorderType eBorderType,
NppStreamContext
nppStreamCtx)
```

Three channel 32-bit floating-point bilateral Gauss filter with border control.

For common parameter descriptions, see [Common parameters for nppiFilterBilateralGaussBorder functions](#).

1.10.5.19 Image Filter High Pass

1.10.5.19.1 FilterHighPass

Filters the image using a high-pass filter kernel.

1.10.5.19.1.1 Common parameters for nppiFilterHighPass functions:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

Enumeration value specifying the mask size.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterHighPass_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions:*

NppStatus **nppiFilterHighPass_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*:

NppStatus **nppiFilterHighPass_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*:

NppStatus **nppiFilterHighPass_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*:

NppStatus **nppiFilterHighPass_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*:

NppStatus **nppiFilterHighPass_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*:

NppStatus **nppiFilterHighPass_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*:

NppStatus **nppiFilterHighPass_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*:

NppStatus **nppiFilterHighPass_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*:

NppStatus **nppiFilterHighPass_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*.

NppStatus **nppiFilterHighPass_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*.

NppStatus **nppiFilterHighPass_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*.

NppStatus **nppiFilterHighPass_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*.

NppStatus **nppiFilterHighPass_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*.

NppStatus **nppiFilterHighPass_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*.

NppStatus **nppiFilterHighPass_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterHighPass functions*.

1.10.5.20 Image Filter High Pass Border

1.10.5.20.1 FilterHighPassBorder

Filters the image using a high-pass filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.5.20.1.1 Common parameters for nppiFilterHighPassBorder functions:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

Enumeration value specifying the mask size.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiFilterHighPassBorder_8u_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus nppiFilterHighPassBorder_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus nppiFilterHighPassBorder_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus nppiFilterHighPassBorder_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus nppiFilterHighPassBorder_16u_C1R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus **nppiFilterHighPassBorder_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus **nppiFilterHighPassBorder_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus **nppiFilterHighPassBorder_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus **nppiFilterHighPassBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus **nppiFilterHighPassBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus **nppiFilterHighPassBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus **nppiFilterHighPassBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus **nppiFilterHighPassBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus **nppiFilterHighPassBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

NppStatus **nppiFilterHighPassBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

```
NppStatus nppiFilterHighPassBorder_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,
NppiSize oSrcSize, NppiPoint oSrcOffset,
Npp32f *pDst, Npp32s nDstStep,
NppiSize oSizeROI, NppiMaskSize
eMaskSize, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterHighPassBorder functions*.

1.10.5.21 Image Filter Low Pass

1.10.5.21.1 FilterLowPass

Filters the image using a low-pass filter kernel.

1.10.5.21.1.1 Common parameters for nppiFilterLowPass functions:

$$\begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} \begin{pmatrix} 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \end{pmatrix}$$

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

Enumeration value specifying the mask size.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiFilterLowPass_8u_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions*.

NppStatus nppiFilterLowPass_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions*.

NppStatus nppiFilterLowPass_8u_C4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions*.

NppStatus nppiFilterLowPass_8u_AC4R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned low-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions*.

NppStatus nppiFilterLowPass_16u_C1R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions*.

NppStatus nppiFilterLowPass_16u_C3R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions*.

NppStatus nppiFilterLowPass_16u_C4R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions*.

NppStatus nppiFilterLowPass_16u_AC4R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned low-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions:*.

NppStatus **nppiFilterLowPass_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions:*.

NppStatus **nppiFilterLowPass_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions:*.

NppStatus **nppiFilterLowPass_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions:*.

NppStatus **nppiFilterLowPass_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed low-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions:*.

NppStatus **nppiFilterLowPass_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions:*.

NppStatus **nppiFilterLowPass_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions:*.

NppStatus **nppiFilterLowPass_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point low-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions:*.

```
NppStatus nppiFilterLowPass_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, Npp32f
                                         *pDst, Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiMaskSize eMaskSize, NppStreamContext
                                         nppStreamCtx)
```

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLowPass functions*.

1.10.5.22 Image Filter Low Pass Border

1.10.5.22.1 FilterLowPassBorder

Filters the image using a low-pass filter kernel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.5.22.1.1 Common parameters for nppiFilterLowPassBorder functions:

$$\begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} \begin{pmatrix} 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \end{pmatrix}$$

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

Enumeration value specifying the mask size.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterLowPassBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus **nppiFilterLowPassBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus nppiFilterLowPassBorder_32f_C4R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point high-pass filter.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

NppStatus nppiFilterLowPassBorder_32f_AC4R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterLowPassBorder functions*.

1.10.5.23 Image Filter Sharpen

1.10.5.23.1 FilterSharpen

Filters the image using a sharpening filter kernel:

1.10.5.23.1.1 Common parameters for nppiFilterSharpen functions:

$$\begin{pmatrix} -1/8 & -1/8 & -1/8 \\ -1/8 & 16/8 & -1/8 \\ -1/8 & -1/8 & -1/8 \end{pmatrix}$$

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterSharpen_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*.

NppStatus **nppiFilterSharpen_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*.

NppStatus **nppiFilterSharpen_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*.

NppStatus **nppiFilterSharpen_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned sharpening filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*.

NppStatus **nppiFilterSharpen_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*.

NppStatus **nppiFilterSharpen_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*.

NppStatus **nppiFilterSharpen_16u_C4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*.

NppStatus **nppiFilterSharpen_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned sharpening filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*:

NppStatus **nppiFilterSharpen_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*:

NppStatus **nppiFilterSharpen_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*:

NppStatus **nppiFilterSharpen_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*:

NppStatus **nppiFilterSharpen_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed sharpening filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*:

NppStatus **nppiFilterSharpen_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*:

NppStatus **nppiFilterSharpen_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*:

NppStatus **nppiFilterSharpen_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point sharpening filter.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*:

NppStatus **nppiFilterSharpen_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point sharpening filter, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSharpen functions*:

1.10.5.24 Image Filter Sharpen Border

1.10.5.24.1 FilterSharpenBorder

Filters the image using a sharpening filter kernel with border control.

If any portion of the 3x3 mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.5.24.1.1 Common parameters for nppiFilterSharpenBorder functions:

$$\begin{pmatrix} -1/8 & -1/8 & -1/8 \\ -1/8 & 16/8 & -1/8 \\ -1/8 & -1/8 & -1/8 \end{pmatrix}$$

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiFilterSharpenBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

NppStatus **nppiFilterSharpenBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

NppStatus **nppiFilterSharpenBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

NppStatus **nppiFilterSharpenBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned sharpening filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

NppStatus **nppiFilterSharpenBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit unsigned sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

NppStatus **nppiFilterSharpenBorder_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit unsigned sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

```
NppStatus nppiFilterSharpenBorder_16u_C4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp16u *pDst, Npp32s nDstStep, NppiSize  
                                              oSizeROI, NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

```
NppStatus nppiFilterSharpenBorder_16u_AC4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp16u *pDst, Npp32s nDstStep, NppiSize  
                                              oSizeROI, NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Four channel 16-bit unsigned sharpening filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

```
NppStatus nppiFilterSharpenBorder_16s_C1R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp16s *pDst, Npp32s nDstStep, NppiSize  
                                              oSizeROI, NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Single channel 16-bit signed sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

```
NppStatus nppiFilterSharpenBorder_16s_C3R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp16s *pDst, Npp32s nDstStep, NppiSize  
                                              oSizeROI, NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Three channel 16-bit signed sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

```
NppStatus nppiFilterSharpenBorder_16s_C4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                              NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                              Npp16s *pDst, Npp32s nDstStep, NppiSize  
                                              oSizeROI, NppiBorderType eBorderType,  
                                              NppStreamContext nppStreamCtx)
```

Four channel 16-bit signed sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

NppStatus nppiFilterSharpenBorder_16s_AC4R_Ctx(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed sharpening filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

NppStatus nppiFilterSharpenBorder_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating-point sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

NppStatus nppiFilterSharpenBorder_32f_C3R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating-point sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

NppStatus nppiFilterSharpenBorder_32f_C4R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point sharpening filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

NppStatus nppiFilterSharpenBorder_32f_AC4R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating-point sharpening filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterSharpenBorder functions*.

1.10.5.25 Image Filter Unsharp Border

1.10.5.25.1 FilterUnsharpBorder

Filters the image using a unsharp-mask sharpening filter kernel with border control.

The algorithm involves the following steps: Smooth the original image with a Gaussian filter, with the width controlled by the `nRadius`. Subtract the smoothed image from the original to create a high-pass filtered image. Apply any clipping needed on the high-pass image, as controlled by the `nThreshold`. Add a certain percentage of the high-pass filtered image to the original image, with the percentage controlled by the `nWeight`. In pseudocode this algorithm can be written as: $\text{HighPass} = \text{Image} - \text{Gaussian}(\text{Image})$ $\text{Result} = \text{Image} + \text{nWeight} * \text{HighPass} * (|\text{HighPass}| \geq \text{nThreshold})$ where `nWeight` is the amount, `nThreshold` is the threshold, and `>=` indicates a Boolean operation, 1 if true, or 0 otherwise.

If any portion of the mask overlaps the source image boundary, the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the `NPP_BORDER_REPLICATE` border type operation is supported.

1.10.5.25.1.1 . Common parameters for `nppiFilterUnsharpBorder` functions:

1.10.5.25.1.2 . Common parameters for `nppiFilterUnsharpGetBufferSize` functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcOffset

The pixel offset that `pSrc` points to relative to the origin of the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nRadius

The radius of the Gaussian filter, in pixels, not counting the center pixel.

param nSigma

The standard deviation of the Gaussian filter, in pixel.

param nWeight

The percentage of the difference between the original and the high pass image that is added back into the original.

param nThreshold

The threshold needed to apply the difference amount.

param eBorderType

The border type operation to be applied at source image border boundaries.

param pDeviceBuffer

Pointer to the user-allocated device scratch buffer required for the unsharp operation.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

param nRadius

The radius of the Gaussian filter, in pixles, not counting the center pixel.

param nSigma

The standard deviation of the Gaussian filter, in pixel.

param hpBufferSize

Pointer to the size of the scratch buffer required for the unsharp operation.

return

Image Data Related Error Codes

Functions

NppStatus **nppiFilterUnsharpBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions:*

NppStatus **nppiFilterUnsharpBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions:*

NppStatus **nppiFilterUnsharpBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions:*.

```
NppStatus nppiFilterUnsharpBorder_8u_AC4R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep,  
                                                NppiPoint oSrcOffset, Npp8u *pDst, Npp32s  
                                                nDstStep, NppiSize oSizeROI, Npp32f  
                                                nRadius, Npp32f nSigma, Npp32f nWeight,  
                                                Npp32f nThreshold, NppiBorderType  
                                                eBorderType, Npp8u *pDeviceBuffer,  
                                                NppStreamContext nppStreamCtx)
```

Four channel 8-bit unsigned unsharp filter (alpha channel is not processed).

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions:*.

```
NppStatus nppiFilterUnsharpBorder_16u_C1R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                                NppiPoint oSrcOffset, Npp16u *pDst,  
                                                Npp32s nDstStep, NppiSize oSizeROI,  
                                                Npp32f nRadius, Npp32f nSigma, Npp32f  
                                                nWeight, Npp32f nThreshold,  
                                                NppiBorderType eBorderType, Npp8u  
                                                *pDeviceBuffer, NppStreamContext  
                                                nppStreamCtx)
```

Single channel 16-bit unsigned unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions:*.

```
NppStatus nppiFilterUnsharpBorder_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                                NppiPoint oSrcOffset, Npp16u *pDst,  
                                                Npp32s nDstStep, NppiSize oSizeROI,  
                                                Npp32f nRadius, Npp32f nSigma, Npp32f  
                                                nWeight, Npp32f nThreshold,  
                                                NppiBorderType eBorderType, Npp8u  
                                                *pDeviceBuffer, NppStreamContext  
                                                nppStreamCtx)
```

Three channel 16-bit unsigned unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions:*.

```
NppStatus nppiFilterUnsharpBorder_16u_C4R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep,  
                                                NppiPoint oSrcOffset, Npp16u *pDst,  
                                                Npp32s nDstStep, NppiSize oSizeROI,  
                                                Npp32f nRadius, Npp32f nSigma, Npp32f  
                                                nWeight, Npp32f nThreshold,  
                                                NppiBorderType eBorderType, Npp8u  
                                                *pDeviceBuffer, NppStreamContext  
                                                nppStreamCtx)
```

Four channel 16-bit unsigned unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions:*.

NppStatus **nppiFilterUnsharpBorder_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four channel 16-bit unsigned unsharp filter (alpha channel is not processed).

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions*:

NppStatus **nppiFilterUnsharpBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions*:

NppStatus **nppiFilterUnsharpBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions*:

NppStatus **nppiFilterUnsharpBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions*:

NppStatus **nppiFilterUnsharpBorder_16s_AC4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed unsharp filter (alpha channel is not processed).

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions*:

NppStatus **nppiFilterUnsharpBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Single channel 32-bit floating point unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions*:

NppStatus **nppiFilterUnsharpBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three channel 32-bit floating point unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions*:

NppStatus **nppiFilterUnsharpBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point unsharp filter.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions*:

NppStatus nppiFilterUnsharpBorder_32f_AC4R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *Npp32f* nRadius, *Npp32f* nSigma, *Npp32f* nWeight, *Npp32f* nThreshold, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four channel 32-bit floating point unsharp filter (alpha channel is not processed).

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpBorder functions:*.

NppStatus nppiFilterUnsharpGetBufferSize_8u_C1R(const *Npp32f* nRadius, const *Npp32f* nSigma, int *hpBufferSize)

Single channel 8-bit unsigned unsharp filter scratch memory size.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpGetBufferSize functions:*.

NppStatus nppiFilterUnsharpGetBufferSize_8u_C3R(const *Npp32f* nRadius, const *Npp32f* nSigma, int *hpBufferSize)

Three channel 8-bit unsigned unsharp filter scratch memory size.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpGetBufferSize functions:*.

NppStatus nppiFilterUnsharpGetBufferSize_8u_C4R(const *Npp32f* nRadius, const *Npp32f* nSigma, int *hpBufferSize)

Four channel 8-bit unsigned unsharp filter scratch memory size.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpGetBufferSize functions:*.

NppStatus nppiFilterUnsharpGetBufferSize_8u_AC4R(const *Npp32f* nRadius, const *Npp32f* nSigma, int *hpBufferSize)

Four channel 8-bit unsigned unsharp filter scratch memory size (alpha channel is not processed).

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpGetBufferSize functions:*.

NppStatus nppiFilterUnsharpGetBufferSize_16u_C1R(const *Npp32f* nRadius, const *Npp32f* nSigma, int *hpBufferSize)

Single channel 16-bit unsigned unsharp filter scratch memory size.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpGetBufferSize functions:*.

NppStatus nppiFilterUnsharpGetBufferSize_16u_C3R(const *Npp32f* nRadius, const *Npp32f* nSigma, int *hpBufferSize)

Three channel 16-bit unsigned unsharp filter scratch memory size.

For common parameter descriptions, see . *Common parameters for nppiFilterUnsharpGetBufferSize functions:*.

NppStatus nppiFilterUnsharpGetBufferSize_16u_C4R(const *Npp32f* nRadius, const *Npp32f* nSigma, int *hpBufferSize)

Four channel 16-bit unsigned unsharp filter scratch memory size.

For common parameter descriptions, see . [Common parameters for nppiFilterUnsharpGetBufferSize functions:](#).

NppStatus nppiFilterUnsharpGetBufferSize_16u_AC4R(const [Npp32f](#) nRadius, const [Npp32f](#) nSigma, int *hpBufferSize)

Four channel 16-bit unsigned unsharp filter scratch memory size (alpha channel is not processed).

For common parameter descriptions, see . [Common parameters for nppiFilterUnsharpGetBufferSize functions:](#).

NppStatus nppiFilterUnsharpGetBufferSize_16s_C1R(const [Npp32f](#) nRadius, const [Npp32f](#) nSigma, int *hpBufferSize)

Single channel 16-bit signed unsharp filter scratch memory size.

For common parameter descriptions, see . [Common parameters for nppiFilterUnsharpGetBufferSize functions:](#).

NppStatus nppiFilterUnsharpGetBufferSize_16s_C3R(const [Npp32f](#) nRadius, const [Npp32f](#) nSigma, int *hpBufferSize)

Three channel 16-bit signed unsharp filter scratch memory size.

For common parameter descriptions, see . [Common parameters for nppiFilterUnsharpGetBufferSize functions:](#).

NppStatus nppiFilterUnsharpGetBufferSize_16s_C4R(const [Npp32f](#) nRadius, const [Npp32f](#) nSigma, int *hpBufferSize)

Four channel 16-bit signed unsharp filter scratch memory size.

For common parameter descriptions, see . [Common parameters for nppiFilterUnsharpGetBufferSize functions:](#).

NppStatus nppiFilterUnsharpGetBufferSize_16s_AC4R(const [Npp32f](#) nRadius, const [Npp32f](#) nSigma, int *hpBufferSize)

Four channel 16-bit signed unsharp filter scratch memory size (alpha channel is not processed).

For common parameter descriptions, see . [Common parameters for nppiFilterUnsharpGetBufferSize functions:](#).

NppStatus nppiFilterUnsharpGetBufferSize_32f_C1R(const [Npp32f](#) nRadius, const [Npp32f](#) nSigma, int *hpBufferSize)

Single channel 32-bit floating point unsharp filter scratch memory size.

For common parameter descriptions, see . [Common parameters for nppiFilterUnsharpGetBufferSize functions:](#).

NppStatus nppiFilterUnsharpGetBufferSize_32f_C3R(const [Npp32f](#) nRadius, const [Npp32f](#) nSigma, int *hpBufferSize)

Three channel 32-bit floating point unsharp filter scratch memory size.

For common parameter descriptions, see . [Common parameters for nppiFilterUnsharpGetBufferSize functions:](#).

NppStatus nppiFilterUnsharpGetBufferSize_32f_C4R(const [Npp32f](#) nRadius, const [Npp32f](#) nSigma, int *hpBufferSize)

Four channel 32-bit floating point unsharp filter scratch memory size.

For common parameter descriptions, see . [Common parameters for nppiFilterUnsharpGetBufferSize functions:](#).

NppStatus **nppiFilterUnsharpGetBufferSize_32f_AC4R**(const *Npp32f* nRadius, const *Npp32f* nSigma, int *hpBufferSize)

Four channel 32-bit floating point unsharp filter scratch memory size (alpha channel is not processed).

For common parameter descriptions, see [. Common parameters for nppiFilterUnsharpGetBufferSize functions:](#)

1.10.5.26 Image Filter Wiener Border

1.10.5.26.1 FilterWienerBorder

Noise removal filtering of an image using an adaptive Wiener filter with border control.

Pixels under the source mask are used to generate statistics about the local neighborhood which are then used to control the amount of adaptive noise filtering locally applied.

Note that if the noise value for a particular channel is set to 0.0f then the output for that channel will contain the square of the variance of local pixels within aMaskSize surrounding each pixel in oSizeROI. Note that is unlikely to be useful unless the pixel data type is floating point due to result clamping. Output from these cases can then be passed through an nppiMean function call using the same oSizeROI. The square root for that channel from the nppiMean call result can then be used as a noise value for a future call to this function if there is no known preexisting noise value.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.5.26.1.1 Common parameters for nppiFilterWienerBorder functions:

For each pixel in the source image the function estimates the local mean and variance in the neighborhood defined by oMaskSize relative to the primary source pixel located at oAnchor.x and oAnchor.y. Given an oMaskSize with width W and height H , the mean, variance, and destination pixel value will be computed per channel as

$$Mean = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

$$Variance^2 = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} (pSrc(j, i)^2 - Mean^2)$$

$$pDst(j, i) = Mean + \frac{(Variance^2 - NoiseVariance^2)}{Variance^2} \cdot (pSrc(j, i) - Mean)$$

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDst*Destination-Image Pointer.***param nDstStep***Destination-Image Line Step.***param oSizeROI***Region-Of-Interest (ROI).***param oMaskSize**

Pixel Width and Height of the rectangular region of interest surrounding the source pixel.

param oAnchor

Positive X and Y relative offsets of primary pixel in region of interest surrounding the source pixel relative to bottom right of oMaskSize.

param aNoise

Fixed size array of per-channel noise variance level value in range of 0.0F to 1.0F.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes*

Functions

NppStatus **nppiFilterWienerBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp32f* aNoise[1], *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 8-bit unsigned Wiener filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

NppStatus **nppiFilterWienerBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp32f* aNoise[3], *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 8-bit unsigned Wiener filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

NppStatus **nppiFilterWienerBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp32f* aNoise[4], *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Wiener filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

NppStatus **nppiFilterWienerBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp32f* aNoise[3], *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 8-bit unsigned Wiener filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

NppStatus **nppiFilterWienerBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp32f* aNoise[1], *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single channel 16-bit signed Wiener filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

NppStatus **nppiFilterWienerBorder_16s_C3R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp32f* aNoise[3], *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three channel 16-bit signed Wiener filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

NppStatus **nppiFilterWienerBorder_16s_C4R_Ctx**(const *Npp16s* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp32f* aNoise[4], *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four channel 16-bit signed Wiener filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

```
NppStatus nppiFilterWienerBorder_16s_AC4R_Ctx(const Npp16s *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp16s *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, NppiSize oMaskSize, NppiPoint  
                                                oAnchor, Npp32f aNoise[3],  
                                                NppiBorderType eBorderType,  
                                                NppStreamContext nppStreamCtx)
```

Four channel 16-bit signed Wiener filter with border control, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

```
NppStatus nppiFilterWienerBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, NppiSize oMaskSize, NppiPoint  
                                                oAnchor, Npp32f aNoise[1], NppiBorderType  
                                                eBorderType, NppStreamContext  
                                                nppStreamCtx)
```

Single channel 32-bit float Wiener filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

```
NppStatus nppiFilterWienerBorder_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, NppiSize oMaskSize, NppiPoint  
                                                oAnchor, Npp32f aNoise[3], NppiBorderType  
                                                eBorderType, NppStreamContext  
                                                nppStreamCtx)
```

Three channel 32-bit float Wiener filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

```
NppStatus nppiFilterWienerBorder_32f_C4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, NppiSize oMaskSize, NppiPoint  
                                                oAnchor, Npp32f aNoise[4], NppiBorderType  
                                                eBorderType, NppStreamContext  
                                                nppStreamCtx)
```

Four channel 32-bit float Wiener filter with border control.

For common parameter descriptions, see *Common parameters for nppiFilterWienerBorder functions*.

```
NppStatus nppiFilterWienerBorder_32f_AC4R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep,  
                                                NppiSize oSrcSize, NppiPoint oSrcOffset,  
                                                Npp32f *pDst, Npp32s nDstStep, NppiSize  
                                                oSizeROI, NppiSize oMaskSize, NppiPoint  
                                                oAnchor, Npp32f aNoise[3],  
                                                NppiBorderType eBorderType,  
                                                NppStreamContext nppStreamCtx)
```

Four channel 32-bit float Wiener filter with border control, ignoring alpha channel.

For common parameter descriptions, see [Common parameters for nppiFilterWienerBorder functions](#):

1.10.5.27 Image Filter Gradient Vector Prewitt Border

1.10.5.27.1 GradientVectorPrewittBorder

RGB Color to Prewitt Gradient Vector conversion using user selected fixed mask size and gradient distance method. Functions support up to 4 optional single channel output gradient vectors, X (vertical), Y (horizontal), magnitude, and angle with user selectable distance methods. Output for a particular vector is disabled by supplying a NULL pointer for that vector. X and Y gradient vectors are in cartesian form in the destination data type.

Magnitude vectors are polar gradient form in the destination data type, angle is always in floating point polar gradient format. Only fixed mask sizes of 3x3 are supported. Only nppiNormL1 (sum) and nppiNormL2 (sqrt of sum of squares) distance methods are currently supported.

Currently only the NPP_BORDER_REPLICATE border type operation is supported. Borderless output can be accomplished by using a larger source image than the destination and adjusting oSrcSize and oSrcOffset parameters accordingly.

1.10.5.27.1.1 Common parameters for nppiFilterGradientVectorPrewittBorder functions:

The following fixed kernel mask is used for producing the pDstX (vertical) output image.

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

The following fixed kernel mask is used for producing the pDstY (horizontal) output image.

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

For the C1R versions of the function the pDstMag output image value for L1 normalization consists of the absolute value of the pDstX value plus the absolute value of the pDstY value at that particular image pixel location. For the C1R versions of the function the pDstMag output image value for L2 normalization consists of the square root of the pDstX value squared plus the pDstY value squared at that particular image pixel location. For the C1R versions of the function the pDstAngle output image value consists of the arctangent (atan2) of the pDstY value and the pDstX value at that particular image pixel location.

For the C3C1R versions of the function, regardless of the selected normalization method, the L2 normalization value is first determined for each of the pDstX and pDstY values for each source channel then the largest L2 normalization value (largest gradient) is used to select which of the 3 pDstX channel values are output to the pDstX image or pDstY channel values are output to the pDstY image. For the C3C1R versions of the function the pDstMag output image value for L1 normalization consists of the same technique used for the C1R version for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 pDstMag channel values to output to the pDstMag image. For the C3C1R versions of the function the pDstMag output image value for L2 normalization consists of just outputting the largest per source channel L2 normalization value to the pDstMag image. For the C3C1R versions of the function the pDstAngle output image value consists of the same technique used for the C1R version calculated for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 angle values to output to the pDstAngle image.

param pSrc*Source-Image Pointer.***param nSrcStep***Source-Image Line Step.***param oSrcSize**

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDstX

X vector destination_image_pointer.

param nDstXStep

X vector destination_image_line_step.

param pDstY

Y vector destination_image_pointer.

param nDstYStep

Y vector destination_image_line_step.

param pDstMag

magnitude destination_image_pointer.

param nDstMagStep

magnitude destination_image_line_step.

param pDstAngle

angle destination_image_pointer.

param nDstAngleStep

angle destination_image_line_step.

param oSizeROI*Region-Of-Interest (ROI).***param eMaskSize**

fixed filter mask size to use.

param eNorm

gradient distance method to use.

param eBorderType

source image border type to use.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiGradientVectorPrewittBorder_8u16s_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDstX, int nDstXStep, *Npp16s* *pDstY, int nDstYStep, *Npp16s* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorPrewittBorder functions*.

NppStatus nppiGradientVectorPrewittBorder_8u16s_C3C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDstX, int nDstXStep, *Npp16s* *pDstY, int nDstYStep, *Npp16s* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorPrewittBorder functions*.

NppStatus **nppiGradientVectorPrewittBorder_16s32f_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorPrewittBorder functions*.

NppStatus **nppiGradientVectorPrewittBorder_16s32f_C3C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorPrewittBorder functions*.

NppStatus **nppiGradientVectorPrewittBorder_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorPrewittBorder functions*.

NppStatus **nppiGradientVectorPrewittBorder_16u32f_C3C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorPrewittBorder functions*.

NppStatus **nppiGradientVectorPrewittBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorPrewittBorder functions*.

NppStatus **nppiGradientVectorPrewittBorder_32f_C3C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorPrewittBorder functions*.

1.10.5.28 Image Filter Gradient Vector Scharr Border

1.10.5.28.1 GradientVectorScharrBorder

RGB Color to Scharr Gradient Vector conversion using user selected fixed mask size and gradient distance method. Functions support up to 4 optional single channel output gradient vectors, X (vertical), Y (horizontal), magnitude, and angle with user selectable distance methods. Output for a particular vector is disabled by supplying a NULL pointer for that vector. X and Y gradient vectors are in cartesian form in the destination data type.

Magnitude vectors are polar gradient form in the destination data type, angle is always in floating

point polar gradient format. Only fixed mask sizes of 3x3 are supported. Only nppiNormL1 (sum) and nppiNormL2 (sqrt of sum of squares) distance methods are currently supported.

Currently only the NPP_BORDER_REPLICATE border type operation is supported. Borderless output can be accomplished by using a larger source image than the destination and adjusting oSrcSize and oSrcOffset parameters accordingly.

1.10.5.28.1.1 Common parameters for nppiFilterGradientVectorScharrBorder functions:

The following fixed kernel mask is used for producing the pDstX (vertical) output image.

$$\begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}$$

The following fixed kernel mask is used for producing the pDstY (horizontal) output image.

$$\begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$$

For the C1R versions of the function the pDstMag output image value for L1 normalization consists of the absolute value of the pDstX value plus the absolute value of the pDstY value at that particular image pixel location. For the C1R versions of the function the pDstMag output image value for L2 normalization consists of the square root of the pDstX value squared plus the pDstY value squared at that particular image pixel location. For the C1R versions of the function the pDstAngle output image value consists of the arctangent (atan2) of the pDstY value and the pDstX value at that particular image pixel location.

For the C3C1R versions of the function, regardless of the selected normalization method, the L2 normalization value is first determined for each of the pDstX and pDstY values for each source channel then the largest L2 normalization value (largest gradient) is used to select which of the 3 pDstX channel values are output to the pDstX image or pDstY channel values are output to the pDstY image. For the C3C1R versions of the function the pDstMag output image value for L1 normalization consists of the same technique used for the C1R version for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 pDstMag channel values to output to the pDstMag image. For the C3C1R versions of the function the pDstMag output image value for L2 normalization consists of just outputting the largest per source channel L2 normalization value to the pDstMag image. For the C3C1R versions of the function the pDstAngle output image value consists of the same technique used for the C1R version calculated for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 angle values to output to the pDstAngle image.

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDstX

X vector destination_image_pointer.

param nDstXStep

X vector destination_image_line_step.

param pDstY

Y vector destination_image_pointer.

param nDstYStep

Y vector destination_image_line_step.

param pDstMag

magnitude destination_image_pointer.

param nDstMagStep

magnitude destination_image_line_step.

param pDstAngle

angle destination_image_pointer.

param nDstAngleStep

angle destination_image_line_step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

fixed filter mask size to use.

param eNorm

gradient distance method to use.

param eBorderType

source image border type to use use.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiGradientVectorScharrBorder_8u16s_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDstX, int nDstXStep, *Npp16s* *pDstY, int nDstYStep, *Npp16s* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorScharrBorder functions*.

NppStatus nppiGradientVectorScharrBorder_8u16s_C3C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDstX, int nDstXStep, *Npp16s* *pDstY, int nDstYStep, *Npp16s* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorScharrBorder functions*.

NppStatus nppiGradientVectorScharrBorder_16s32f_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see [Common parameters for nppiFilterGradientVectorScharBorder functions](#).

```
NppStatus nppiGradientVectorScharBorder_16s32f_C3C1R_Ctx( const Npp16s *pSrc, int
                                                         nSrcStep, NppiSize
                                                         oSrcSize, NppiPoint
                                                         oSrcOffset, Npp32f *pDstX,
                                                         int nDstXStep, Npp32f
                                                         *pDstY, int nDstYStep,
                                                         Npp32f *pDstMag, int
                                                         nDstMagStep, Npp32f
                                                         *pDstAngle, int
                                                         nDstAngleStep, NppiSize
                                                         oSizeROI, NppiMaskSize
                                                         eMaskSize, NppiNorm
                                                         eNorm, NppiBorderType
                                                         eBorderType,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

3 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see [Common parameters for nppiFilterGradientVectorScharBorder functions](#).

```
NppStatus nppiGradientVectorScharBorder_16u32f_C1R_Ctx( const Npp16u *pSrc, int
                                                         nSrcStep, NppiSize oSrcSize,
                                                         NppiPoint oSrcOffset, Npp32f
                                                         *pDstX, int nDstXStep,
                                                         Npp32f *pDstY, int nDstYStep,
                                                         Npp32f *pDstMag, int
                                                         nDstMagStep, Npp32f
                                                         *pDstAngle, int
                                                         nDstAngleStep, NppiSize
                                                         oSizeROI, NppiMaskSize
                                                         eMaskSize, NppiNorm eNorm,
                                                         NppiBorderType eBorderType,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

1 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see [Common parameters for nppiFilterGradientVectorScharBorder functions](#).

NppStatus nppiGradientVectorScharrBorder_16u32f_C3C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorScharrBorder functions*.

NppStatus nppiGradientVectorScharrBorder_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorScharrBorder functions*.

NppStatus nppiGradientVectorScharrBorder_32f_C3C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see [Common parameters for nppiFilterGradientVectorSobelBorder functions](#):

1.10.5.29 Image Filter Gradient Vector Sobel Border

1.10.5.29.1 GradientVectorSobelBorder

RGB Color to Sobel Gradient Vector conversion using user selected fixed mask size and gradient distance method. Functions support up to 4 optional single channel output gradient vectors, X (vertical), Y (horizontal), magnitude, and angle with user selectable distance methods. Output for a particular vector is disabled by supplying a NULL pointer for that vector. X and Y gradient vectors are in cartesian form in the destination data type.

Magnitude vectors are polar gradient form in the destination data type, angle is always in floating point polar gradient format. Only fixed mask sizes of 3x3 and 5x5 are supported. Only nppiNormL1 (sum) and nppiNormL2 (sqrt of sum of squares) distance methods are currently supported.

Currently only the NPP_BORDER_REPLICATE border type operation is supported. Borderless output can be accomplished by using a larger source image than the destination and adjusting oSrcSize and oSrcOffset parameters accordingly.

1.10.5.29.1.1 Common parameters for nppiFilterGradientVectorSobelBorder functions:

One of the following fixed kernel masks are used for producing the 3x3 or 5x5 pDstX (vertical) output image depending on selected mask size.

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{pmatrix}$$

One of the following fixed kernel masks are used for producing the 3x3 or 5x5 pDstY (horizontal) output image depending on selected mask size.

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{pmatrix}$$

For the C1R versions of the function the pDstMag output image value for L1 normalization consists of the absolute value of the pDstX value plus the absolute value of the pDstY value at that particular image pixel location. For the C1R versions of the function the pDstMag output image value for L2 normalization consists of the square root of the pDstX value squared plus the pDstY value squared at that particular image pixel location. For the C1R versions of the function the pDstAngle output image value consists of the arctangent (atan2) of the pDstY value and the pDstX value at that particular image pixel location.

For the C3C1R versions of the function, regardless of the selected normalization method, the L2 normalization value is first determined for each of the pDstX and pDstY values for each source channel then the largest L2 normalization value (largest gradient) is used to select which of the 3 pDstX channel values are output to the pDstX image or pDstY channel values are output to the pDstY image. For the C3C1R versions of the function the pDstMag output image value for L1 normalization consists of the same technique used for the C1R version for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 pDstMag channel values to output to the pDstMag image. For the C3C1R versions of the function the pDstMag output image value for L2 normalization consists of just outputting the largest per source channel L2 normalization value to the pDstMag image. For the C3C1R versions of the function the pDstAngle output image value consists of the same technique used for the C1R version calculated for each source image channel. Then the largest L2 normalization value is again used to select which of the 3 angle values to output to the pDstAngle image.

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param pDstX

X vector destination_image_pointer.

param nDstXStep

X vector destination_image_line_step.

param pDstY

Y vector destination_image_pointer.

param nDstYStep

Y vector destination_image_line_step.

param pDstMag

magnitude destination_image_pointer.

param nDstMagStep

magnitude destination_image_line_step.

param pDstAngle

angle destination_image_pointer.

param nDstAngleStep

angle destination_image_line_step.

param oSizeROI

Region-Of-Interest (ROI).

param eMaskSize

fixed filter mask size to use.

param eNorm

gradient distance method to use.

param eBorderType

source image border type to use use.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiGradientVectorSobelBorder_8u16s_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDstX, int nDstXStep, *Npp16s* *pDstY, int nDstYStep, *Npp16s* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorSobel-Border functions*.


```
NppStatus nppiGradientVectorSobelBorder_8u16s_C3C1R_Ctx(const Npp8u *pSrc, int
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16s
*pDstX, int nDstXStep,
Npp16s *pDstY, int nDstYStep,
Npp16s *pDstMag, int
nDstMagStep, Npp32f
*pDstAngle, int
nDstAngleStep, NppiSize
oSizeROI, NppiMaskSize
eMaskSize, NppiNorm eNorm,
NppiBorderType eBorderType,
NppStreamContext
nppStreamCtx)
```

3 channel 8-bit unsigned packed RGB to optional 1 channel 16-bit signed X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorSobelBorder functions*.

```
NppStatus nppiGradientVectorSobelBorder_16s32f_C1R_Ctx(const Npp16s *pSrc, int
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f
*pDstX, int nDstXStep, Npp32f
*pDstY, int nDstYStep, Npp32f
*pDstMag, int nDstMagStep,
Npp32f *pDstAngle, int
nDstAngleStep, NppiSize
oSizeROI, NppiMaskSize
eMaskSize, NppiNorm eNorm,
NppiBorderType eBorderType,
NppStreamContext
nppStreamCtx)
```

1 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorSobelBorder functions*.

```
NppStatus nppiGradientVectorSobelBorder_16s32f_C3C1R_Ctx(const Npp16s *pSrc, int
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset,
Npp32f *pDstX, int
nDstXStep, Npp32f *pDstY,
int nDstYStep, Npp32f
*pDstMag, int nDstMagStep,
Npp32f *pDstAngle, int
nDstAngleStep, NppiSize
oSizeROI, NppiMaskSize
eMaskSize, NppiNorm
eNorm, NppiBorderType
eBorderType,
NppStreamContext
nppStreamCtx)
```

3 channel 16-bit signed packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

```
NppStatus nppiGradientVectorSobelBorder_16u32f_C1R_Ctx(const Npp16u *pSrc, int
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f
*pDstX, int nDstXStep, Npp32f
*pDstY, int nDstYStep, Npp32f
*pDstMag, int nDstMagStep,
Npp32f *pDstAngle, int
nDstAngleStep, NppiSize
oSizeROI, NppiMaskSize
eMaskSize, NppiNorm eNorm,
NppiBorderType eBorderType,
NppStreamContext
nppStreamCtx)
```

1 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorSobel-Border functions*.

```
NppStatus nppiGradientVectorSobelBorder_16u32f_C3C1R_Ctx(const Npp16u *pSrc, int
nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset,
Npp32f *pDstX, int
nDstXStep, Npp32f *pDstY,
int nDstYStep, Npp32f
*pDstMag, int nDstMagStep,
Npp32f *pDstAngle, int
nDstAngleStep, NppiSize
oSizeROI, NppiMaskSize
eMaskSize, NppiNorm
eNorm, NppiBorderType
eBorderType,
NppStreamContext
nppStreamCtx)
```

3 channel 16-bit unsigned packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorSobel-Border functions*.

NppStatus **nppiGradientVectorSobelBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorSobel-Border functions*.

NppStatus **nppiGradientVectorSobelBorder_32f_C3C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDstX, int nDstXStep, *Npp32f* *pDstY, int nDstYStep, *Npp32f* *pDstMag, int nDstMagStep, *Npp32f* *pDstAngle, int nDstAngleStep, *NppiSize* oSizeROI, *NppiMaskSize* eMaskSize, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point packed RGB to optional 1 channel 32-bit floating point X (vertical), Y (horizontal), magnitude, and/or 32-bit floating point angle gradient vectors with user selectable fixed mask size and distance method with border control.

For common parameter descriptions, see *Common parameters for nppiFilterGradientVectorSobel-Border functions*.

1.10.6. Computer Vision Filtering Functions

1.10.6.1 Computer Vision

The set of computer vision functions available in the library.

1.10.6.2 Image Filter Distance Transform

1.10.6.2.1 FilterDistanceTransform

Performs Exact Euclidean Distance Transform function using the Parallel Banding Algorithm (PBA+) defined by Tiow-Seng Tan, et al paper named “Parallel Banding Algorithm to Compute Exact Distance Transform with the GPU” published dated August 8, 2019.

Output for these functions is an optional 16-bit signed integer voronoi diagram (pairs of signed 16 bit integer x, y distance values) and/or an optional true euclidean distance transform image generated from the internal voronoi diagram in either unsigned 16-bit truncated integer format or 32-bit floating point format. Additional optional output can include an signed 16-bit integer Voronoi diagram containing site indices and/or a signed 16-bit integer Voronoi diagram containing relative Manhattan distances to the closest sites. Minimum and maximum image ROI widths and heights are 64 and 32767.

Note that an input image that does not contain at least one site pixel is considered to be an invalid image. If you suspect that your input image may be invalid you can call an NPP function like `nppi_CountInRange()` first to confirm that the image is valid before calling the distance transform function.

The `nMinSiteValue` and `nMaxSiteValue` parameters can be used to control which source image pixels are considered sites (traditionally 0) and non-sites (everything else).

Antialiased true distance transform, when available, is only available as double precision floating point (`Npp64f`) output data only and is enabled by setting the `pAntialiasingDeviceBuffer` pointer parameter to a non-NULL value.

The algorithm used for antialiasing is derived from the `edtaa4` version from “Anti-aliased Euclidean distance transform” by Stefan Gustavson et. al. published in 2009 and is used under the permissions specified below.

Derived from `edtaa4.c` - compute the Euclidean distance transform of an image, with more accurate handling of 1 pixel wide anti-aliased edges.

This is a MEX-file for MATLAB. MATLAB is a product of The MathWorks, Inc.

Code in “`edtaa4func.c`” originally by Stefan Gustavson 1994, implemented from a verbal description in the PhD dissertation of Ingemar Ragnemalm, dept of EE, Linköping University.

Modification to handle antialiased edges and this Matlab MEX wrapper by Stefan Gustavson, (stefan.gustavson@gmail.com) 2009-05-17

Copyright (C) 2009 Stefan Gustavson (stefan.gustavson@gmail.com)

This software is distributed under the permissive “MIT License”:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Functions

NppStatus **nppiDistanceTransformPBAGetBufferSize**(*NppiSize* oSizeROI, size_t *hpBufferSize)

Calculate scratch buffer size needed for the DistanceTransformPBA function based on destination image SizeROI width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBAGetAntialiasingBufferSize**(*NppiSize* oSizeROI, size_t *hpAntialiasingBufferSize)

Calculate scratch buffer size needed for the DistanceTransformPBA function antialiasing based on destination image SizeROI width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpAntialiasingBufferSize** – Optional buffer size. Important: hpAntialiasingBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSignedDistanceTransformPBAGetBufferSize**(*NppiSize* oSizeROI, size_t *hpBufferSize)

Calculate scratch buffer size needed for the DistanceTransformPBA Antialiasing function based on destination image SizeROI width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSignedDistanceTransformPBAGet64fBufferSize**(*NppiSize* oSizeROI, size_t *hpBufferSize)

Calculate scratch buffer size needed for the SignedDistanceTransformPBA function when transform output data type is Npp64f based on destination image SizeROI width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSignedDistanceTransformPBA**GetAntialiasingBufferSize(*NppiSize* oSizeROI, size_t *hpAntialiasingBufferSize)

Calculate scratch buffer size needed for the SignedDistanceTransformPBA function antialiasing based on destination image SizeROI width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpAntialiasingBufferSize** – Optional buffer size. Important: hpAntialiasingBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_8u16u_C1R_Ctx**(*Npp8u* *pSrc, int nSrcStep, *Npp8u* nMinSiteValue, *Npp8u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp16u* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional unsigned 16-bit truncated integer transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.

- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp16u)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_8u16u_C1R_Ctx**(*Npp8u* *pSrc, int nSrcStep, *Npp8u* nMinSiteValue, *Npp8u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp16u* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional unsigned 16-bit truncated integer transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).

- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp16u)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_8s16u_C1R_Ctx**(*Npp8s* *pSrc, int nSrcStep, *Npp8s* nMinSiteValue, *Npp8s* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp16u* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional unsigned 16-bit truncated integer transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.

- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp16u)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_8s16u_C1R_Ctx**(*Npp8s* *pSrc, int nSrcStep, *Npp8s* nMinSiteValue, *Npp8s* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp16u* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional unsigned 16-bit truncated integer transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).

- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof(Npp16s)}$).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof(Npp16s)}$).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof(Npp16s)}$).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least $\text{oSizeROI.width} * \text{sizeof(Npp16u)}$).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size `hpBufferSize` (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_16u16u_C1R_Ctx**(*Npp16u* *pSrc, int nSrcStep, *Npp16u* nMinSiteValue, *Npp16u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp16u* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional unsigned 16-bit truncated integer transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq \text{nMinSiteValue}$ and $\leq \text{nMaxSiteValue}$ are considered sites (traditionally 0s).

- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof(Npp16s)}$).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof(Npp16s)}$).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof(Npp16s)}$).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least $\text{oSizeROI.width} * \text{sizeof(Npp16u)}$).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_16u16u_C1R_Ctx**(*Npp16u* *pSrc, int nSrcStep, *Npp16u* nMinSiteValue, *Npp16u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp16u* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional unsigned 16-bit truncated integer transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp16u)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_16s16u_C1R_Ctx**(*Npp16s* *pSrc, int nSrcStep, *Npp16s* nMinSiteValue, *Npp16s* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp16u* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional unsigned 16-bit truncated integer transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp16u)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```

NppStatus nppiDistanceTransformAbsPBA_16s16u_C1R_Ctx(Npp16s *pSrc, int nSrcStep,
                                                    Npp16s nMinSiteValue, Npp16s
                                                    nMaxSiteValue, Npp16s
                                                    *pDstVoronoi, int
                                                    nDstVoronoiStep, Npp16s
                                                    *pDstVoronoiIndices, int
                                                    nDstVoronoiIndicesStep, Npp16u
                                                    *pDstVoronoiAbsoluteManhattan-
                                                    Distances, int
                                                    nDstVoronoiAbsoluteManhattan-
                                                    DistancesStep, Npp16u
                                                    *pDstTransform, int
                                                    nDstTransformStep, NppiSize
                                                    oSizeROI, Npp8u *pDeviceBuffer,
                                                    NppStreamContext
                                                    nppStreamCtx)

```

1 channel 16-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional unsigned 16-bit truncated integer transform with absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp16u)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBuffer-Size (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiDistanceTransformPBA_8u32f_C1R_Ctx**(*Npp8u* *pSrc, int nSrcStep, *Npp8u* nMinSiteValue, *Npp8u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp32f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 32-bit floating point transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBuffer-Size (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_8u32f_C1R_Ctx**(*Npp8u* *pSrc, int nSrcStep, *Npp8u* nMinSiteValue, *Npp8u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp32f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 32-bit floating point transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.

- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_8s32f_C1R_Ctx**(*Npp8s* *pSrc, int nSrcStep, *Npp8s* nMinSiteValue, *Npp8s* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp32f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 32-bit floating point transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).

- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_8s32f_C1R_Ctx**(*Npp8s* *pSrc, int nSrcStep, *Npp8s* nMinSiteValue, *Npp8s* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp32f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 32-bit floating point transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.

- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_16u32f_C1R_Ctx**(*Npp16u* *pSrc, int nSrcStep, *Npp16u* nMinSiteValue, *Npp16u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp32f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 32-bit floating point transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).

- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_16u32f_C1R_Ctx**(*Npp16u* *pSrc, int nSrcStep, *Npp16u* nMinSiteValue, *Npp16u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp32f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 32-bit floating point transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).

- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBuffer-Size (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_16s32f_C1R_Ctx**(*Npp16s* *pSrc, int nSrcStep, *Npp16s* nMinSiteValue, *Npp16s* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp32f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 32-bit floating point transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.

- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_16s32f_C1R_Ctx**(*Npp16s* *pSrc, int nSrcStep, *Npp16s* nMinSiteValue, *Npp16s* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp32f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 32-bit floating point transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).

- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_8u64f_C1R_Ctx**(*Npp8u* *pSrc, int nSrcStep, *Npp8u* nMinSiteValue, *Npp8u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.

- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof}(\text{Npp16s})$).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof}(\text{Npp16s})$).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof}(\text{Npp16s})$).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least $\text{oSizeROI.width} * \text{sizeof}(\text{Npp64f})$).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_8u64f_C1R_Ctx**(*Npp8u* *pSrc, int nSrcStep, *Npp8u* nMinSiteValue, *Npp8u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance

voronoi diagram output and/or optional 64-bit floating point transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp64f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```

NppStatus nppiDistanceTransformPBA_8s64f_C1R_Ctx(Npp8s *pSrc, int nSrcStep, Npp8s
nMinSiteValue, Npp8s nMaxSiteValue,
Npp16s *pDstVoronoi, int
nDstVoronoiStep, Npp16s
*pDstVoronoiIndices, int
nDstVoronoiIndicesStep, Npp16s
*pDstVoronoiRelativeManhattanDis-
tances, int
nDstVoronoiRelativeManhattanDis-
tancesStep, Npp64f *pDstTransform,
int nDstTransformStep, NppiSize
oSizeROI, Npp8u *pDeviceBuffer, Npp8u
*pAntialiasingDeviceBuffer,
NppStreamContext nppStreamCtx)

```

1 channel 8-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp64f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBuffer-Size (see *nppiDistanceTransformPBAGetBufferSize()* above)

- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_8s64f_C1R_Ctx**(*Npp8s* *pSrc, int nSrcStep, *Npp8s* nMinSiteValue, *Npp8s* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).

- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp64f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiDistanceTransformPBA_16u64f_C1R_Ctx**(*Npp16u* *pSrc, int nSrcStep, *Npp16u* nMinSiteValue, *Npp16u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).

- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp64f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_16u64f_C1R_Ctx**(*Npp16u* *pSrc, int nSrcStep, *Npp16u* nMinSiteValue, *Npp16u* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).

- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof(Npp16s)}$).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof(Npp16s)}$).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof(Npp16s)}$).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least $\text{oSizeROI.width} * \text{sizeof(Npp64f)}$).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_16s64f_C1R_Ctx**(*Npp16s* *pSrc, int nSrcStep, *Npp16s* nMinSiteValue, *Npp16s* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.

- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp64f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes


```

NppStatus nppiDistanceTransformAbsPBA_16s64f_C1R_Ctx(Npp16s *pSrc, int nSrcStep,
                                                    Npp16s nMinSiteValue, Npp16s
                                                    nMaxSiteValue, Npp16s
                                                    *pDstVoronoi, int
                                                    nDstVoronoiStep, Npp16s
                                                    *pDstVoronoiIndices, int
                                                    nDstVoronoiIndicesStep, Npp16u
                                                    *pDstVoronoiAbsoluteManhattan-
                                                    Distances, int
                                                    nDstVoronoiAbsoluteManhattan-
                                                    DistancesStep, Npp64f
                                                    *pDstTransform, int
                                                    nDstTransformStep, NppiSize
                                                    oSizeROI, Npp8u *pDeviceBuffer,
                                                    Npp8u *pAntialiasingDeviceBuffer,
                                                    NppStreamContext
                                                    nppStreamCtx)

```

1 channel 16-bit signed grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – signed source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp64f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_32f64f_C1R_Ctx**(*Npp32f* *pSrc, int nSrcStep, *Npp32f* nMinSiteValue, *Npp32f* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).

- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp64f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_32f64f_C1R_Ctx**(*Npp32f* *pSrc, int nSrcStep, *Npp32f* nMinSiteValue, *Npp32f* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).

- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp64f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformPBA_64f_C1R_Ctx**(*Npp64f* *pSrc, int nSrcStep, *Npp64f* nMinSiteValue, *Npp64f* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16s* *pDstVoronoiRelativeManhattanDistances, int nDstVoronoiRelativeManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 64-bit floating point grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional relative Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).

- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof}(\text{Npp16s})$).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof}(\text{Npp16s})$).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least $\text{oSizeROI.width} * 2 * \text{sizeof}(\text{Npp16s})$).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least $\text{oSizeROI.width} * \text{sizeof}(\text{Npp64f})$).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiDistanceTransformAbsPBA_64f_C1R_Ctx**(*Npp64f* *pSrc, int nSrcStep, *Npp64f* nMinSiteValue, *Npp64f* nMaxSiteValue, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 64-bit floating point grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram output and/or optional 64-bit floating point transform with optional absolute Manhattan distances.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nMinSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **nMaxSiteValue** – source image pixel values $\geq$ nMinSiteValue and $\leq$ nMaxSiteValue are considered sites (traditionally 0s).
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp64f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiDistanceTransformPBAGetBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiSignedDistanceTransformPBA_32f_C1R_Ctx(Npp32f *pSrc, int nSrcStep,  
                                                    Npp32f nCutoffValue, Npp32f  
                                                    nSubPixelXShift, Npp32f  
                                                    nSubPixelYShift, Npp16s  
                                                    *pDstVoronoi, int  
                                                    nDstVoronoiStep, Npp16s  
                                                    *pDstVoronoiIndices, int  
                                                    nDstVoronoiIndicesStep, Npp16s  
                                                    *pDstVoronoiRelativeManhattan-  
Distances, int  
                                                    nDstVoronoiRelativeManhattan-  
DistancesStep, Npp32f  
                                                    *pDstTransform, int  
                                                    nDstTransformStep, NppiSize  
oSizeROI, Npp8u *pDeviceBuffer,  
NppStreamContext  
nppStreamCtx)
```

1 channel 32-bit floating point grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram and 32-bit floating point transform with optional sub-pixel shifts.

For this particular version of the function acceptable input pixel intensities are less than or equal to 0.0f for those fully outside of connected pixel regions, intensities with fractional parts between 0.0f and 1.0f representing the percentage of connected pixel region sub-pixel coverage within a particular pixel (region contour), and intensities greater than or equal to 1.0f for pixels that are fully contained within closed connected pixel regions. This function executes in two passes, the first pass prioritizes pixels outside of closed regions, the second pass prioritizes pixels within closed regions. The two passes are then merged on output. The function assumes that fully covered pixels have centers located at sub-pixel locations of .5,.5 . In general, object exterior distances are output as negative numbers and object interior distances are output as positive numbers.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nCutoffValue** – source image pixel values < nCutoffValue will be considered fully outside of pixel regions (and set to -1).
- ▶ **nSubPixelXShift** – final transform distances will be shifted in the X direction by this sub-pixel fraction.
- ▶ **nSubPixelYShift** – final transform distances will be shifted in the Y direction by this sub-pixel fraction.
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).

- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiSignedDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSignedDistanceTransformAbsPBA_32f_C1R_Ctx**(*Npp32f* *pSrc, int nSrcStep, *Npp32f* nCutoffValue, *Npp32f* nSubPixelXShift, *Npp32f* nSubPixelYShift, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp32f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram and 32-bit floating point transform with optional sub-pixel shifts.

For this particular version of the function acceptable input pixel intensities are less than or equal to 0.0f for those fully outside of connected pixel regions, intensities with fractional parts between 0.0f and 1.0f representing the percentage of connected pixel region sub-pixel coverage within a particular pixel (region contour), and intensities greater than or equal to 1.0f for pixels that are fully contained within closed connected pixel regions. This function executes in two passes, the first pass prioritizes pixels outside of closed regions, the second pass prioritizes pixels within closed regions. The two passes are then merged on output. The function assumes that fully covered pixels have centers located at sub-pixel locations of .5,.5 . In general, object exterior distances are output as negative numbers progressing to positive and object interior distances are output as positive numbers progressing to negative.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.

- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nCutoffValue** – source image pixel values < nCutoffValue will be considered fully outside of pixel regions (and set to -1).
- ▶ **nSubPixelXShift** – final transform distances will be shifted in the X direction by this sub-pixel fraction.
- ▶ **nSubPixelYShift** – final transform distances will be shifted in the Y direction by this sub-pixel fraction.
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiSignedDistanceTransformPBAGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes


```

NppStatus nppiSignedDistanceTransformPBA_32f64f_C1R_Ctx(
    Npp32f *pSrc, int nSrcStep,
    Npp32f nCutoffValue, Npp64f
    nSubPixelXShift, Npp64f
    nSubPixelYShift, Npp16s
    *pDstVoronoi, int
    nDstVoronoiStep, Npp16s
    *pDstVoronoiIndices, int
    nDstVoronoiIndicesStep,
    Npp16s *pDstVoronoiRelative-
    ManhattanDistances, int
    nDstVoronoiRelativeManhat-
    tanDistancesStep, Npp64f
    *pDstTransform, int
    nDstTransformStep, NppiSize
    oSizeROI, Npp8u
    *pDeviceBuffer, Npp8u
    *pAntialiasingDeviceBuffer,
    NppStreamContext
    nppStreamCtx)

```

1 channel 32-bit floating point grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram and 64-bit floating point transform with optional sub-pixel shifts.

For this particular version of the function acceptable input pixel intensities are less than or equal to 0.0f for those fully outside of connected pixel regions, intensities with fractional parts between 0.0f and 1.0f representing the percentage of connected pixel region sub-pixel coverage within a particular pixel (region contour), and intensities greater than or equal to 1.0f for pixels that are fully contained within closed connected pixel regions. This function executes in two passes, the first pass prioritizes pixels outside of closed regions, the second pass prioritizes pixels within closed regions. The two passes are then merged on output. The function assumes that fully covered pixels have centers located at sub-pixel locations of .5,.5 . In general, object exterior distances are output as negative numbers progressing to positive and object interior distances are output as positive numbers progressing to negative.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **nCutoffValue** – source image pixel values < nCutoffValue will be considered fully outside of pixel regions (and set to -1).
- **nSubPixelXShift** – final transform distances will be shifted in the X direction by this sub-pixel fraction.
- **nSubPixelYShift** – final transform distances will be shifted in the Y direction by this sub-pixel fraction.
- **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.

- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiSignedDistanceTransformPBAGet64fBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiSignedDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSignedDistanceTransformAbsPBA_32f64f_C1R_Ctx**(*Npp32f* *pSrc, int nSrcStep, *Npp32f* nCutoffValue, *Npp64f* nSubPixelXShift, *Npp64f* nSubPixelYShift, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram and 64-bit floating point transform with optional sub-pixel shifts.

For this particular version of the function acceptable input pixel intensities are less than or equal to 0.0f for those fully outside of connected pixel regions, intensities with fractional parts between 0.0f and 1.0f representing the percentage of connected pixel region sub-pixel coverage within a particular pixel (region contour), and intensities greater than or equal to 1.0f for pixels that are fully contained within closed connected pixel regions. This function executes in two passes, the

first pass prioritizes pixels outside of closed regions, the second pass prioritizes pixels within closed regions. The two passes are then merged on output. The function assumes that fully covered pixels have centers located at sub-pixel locations of .5,.5 . In general, object exterior distances are output as negative numbers progressing to positive and object interior distances are output as positive numbers progressing to negative.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **nCutoffValue** – source image pixel values < nCutoffValue will be considered fully outside of pixel regions (and set to -1).
- ▶ **nSubPixelXShift** – final transform distances will be shifted in the X direction by this sub-pixel fraction.
- ▶ **nSubPixelYShift** – final transform distances will be shifted in the Y direction by this sub-pixel fraction.
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiSignedDistanceTransformPBAGet64fBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiSignedDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

```

NppStatus nppiSignedDistanceTransformPBA_64f_C1R_Ctx(Npp64f *pSrc, int nSrcStep,
Npp64f nCutoffValue, Npp64f
nSubPixelXShift, Npp64f
nSubPixelYShift, Npp16s
*pDstVoronoi, int
nDstVoronoiStep, Npp16s
*pDstVoronoiIndices, int
nDstVoronoiIndicesStep, Npp16s
*pDstVoronoiRelativeManhattan-
Distances, int
nDstVoronoiRelativeManhattan-
DistancesStep, Npp64f
*pDstTransform, int
nDstTransformStep, NppiSize
oSizeROI, Npp8u *pDeviceBuffer,
Npp8u *pAntialiasingDeviceBuffer,
NppStreamContext
nppStreamCtx)

```

1 channel 64-bit floating point grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram and 64-bit floating point transform with optional sub-pixel shifts.

For this particular version of the function acceptable input pixel intensities are less than or equal to 0.0f for those fully outside of connected pixel regions, intensities with fractional parts between 0.0f and 1.0f representing the percentage of connected pixel region sub-pixel coverage within a particular pixel (region contour), and intensities greater than or equal to 1.0f for pixels that are fully contained within closed connected pixel regions. This function executes in two passes, the first pass prioritizes pixels outside of closed regions, the second pass prioritizes pixels within closed regions. The two passes are then merged on output. The function assumes that fully covered pixels have centers located at sub-pixel locations of .5,.5 . In general, object exterior distances are output as negative numbers progressing to positive and object interior distances are output as positive numbers progressing to negative.

Parameters

- **pSrc** – *Source-Image Pointer*.
- **nSrcStep** – *Source-Image Line Step*.
- **nCutoffValue** – source image pixel values < nCutoffValue will be considered fully outside of pixel regions (and set to -1).
- **nSubPixelXShift** – final transform distances will be shifted in the X direction by this sub-pixel fraction.
- **nSubPixelYShift** – final transform distances will be shifted in the Y direction by this sub-pixel fraction.
- **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).

- ▶ **pDstVoronoiRelativeManhattanDistances** – device memory voronoi relative Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiRelativeManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiSignedDistanceTransformPBAGet64fBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSignedDistanceTransformAbsPBA_64f_C1R_Ctx**(*Npp64f* *pSrc, int nSrcStep, *Npp64f* nCutoffValue, *Npp64f* nSubPixelXShift, *Npp64f* nSubPixelYShift, *Npp16s* *pDstVoronoi, int nDstVoronoiStep, *Npp16s* *pDstVoronoiIndices, int nDstVoronoiIndicesStep, *Npp16u* *pDstVoronoiAbsoluteManhattanDistances, int nDstVoronoiAbsoluteManhattanDistancesStep, *Npp64f* *pDstTransform, int nDstTransformStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pAntialiasingDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 64-bit floating point grayscale to optional 1 channel 16-bit signed integer euclidean distance voronoi diagram and 64-bit floating point transform with optional sub-pixel shifts.

For this particular version of the function acceptable input pixel intensities are less than or equal to 0.0f for those fully outside of connected pixel regions, intensities with fractional parts between 0.0f and 1.0f representing the percentage of connected pixel region sub-pixel coverage within a particular pixel (region contour), and intensities greater than or equal to 1.0f for pixels that are fully contained within closed connected pixel regions. This function executes in two passes, the first pass prioritizes pixels outside of closed regions, the second pass prioritizes pixels within closed regions. The two passes are then merged on output. The function assumes that fully covered pixels have centers located at sub-pixel locations of .5,.5 . In general, object exterior distances are output as negative numbers progressing to positive and object interior distances are output as positive numbers progressing to negative.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **nCutoffValue** – source image pixel values < nCutoffValue will be considered fully outside of pixel regions (and set to -1).
- ▶ **nSubPixelXShift** – final transform distances will be shifted in the X direction by this sub-pixel fraction.
- ▶ **nSubPixelYShift** – final transform distances will be shifted in the Y direction by this sub-pixel fraction.
- ▶ **pDstVoronoi** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi output.
- ▶ **nDstVoronoiStep** – voronoi destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiIndices** – device memory voronoi diagram destination_image_pointer or NULL for no voronoi indices output.
- ▶ **nDstVoronoiIndicesStep** – voronoi indices destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstVoronoiAbsoluteManhattanDistances** – device memory voronoi absolute Manhattan distances destination_image_pointer or NULL for no voronoi Manhattan output.
- ▶ **nDstVoronoiAbsoluteManhattanDistancesStep** – voronoi Manhattan destination_image_line_step (must be at least oSizeROI.width * 2 * sizeof(Npp16s)).
- ▶ **pDstTransform** – device memory true euclidean distance transform destination_image_pointer or NULL for no transform output.
- ▶ **nDstTransformStep** – true euclidean distance transform destination_image_line_step (must be at least oSizeROI.width * sizeof(Npp32f)).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBufferSize (see *nppiSignedDistanceTransformPBAGet64fBufferSize()* above)
- ▶ **pAntialiasingDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpAntialiasingBufferSize (see *nppiDistanceTransformPBAGetAntialiasingBufferSize()* above) or NULL if not Antialiasing
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.6.3 Image Filter Canny Border

1.10.6.3.1 FilterCannyBorder

Performs Canny edge detection on a single channel 8-bit grayscale image and outputs a single channel 8-bit image consisting of 0x00 and 0xFF values with 0xFF representing edge pixels.

The algorithm consists of three phases. The first phase generates two output images consisting of a single channel 16-bit signed image containing magnitude values and a single channel 32-bit floating point image containing the angular direction of those magnitude values. This phase is accomplished by calling the appropriate GradientVectorBorder filter function based on the filter type, filter mask size, and norm type requested. The next phase uses those magnitude and direction images to suppress non-maximum magnitude values which are lower than the values of either of its two nearest neighbors in the same direction as the test magnitude pixel in the 3x3 surrounding magnitude pixel neighborhood. This phase outputs a new magnitude image with non-maximum pixel values suppressed. Finally, in the third phase, the new magnitude image is passed through a hysteresis threshold filter that filters out any magnitude values that are not connected to another edge magnitude value. In this phase, any magnitude value above the high threshold value is automatically accepted, any magnitude value below the low threshold value is automatically rejected. For magnitude values that lie between the low and high threshold, values are only accepted if one of their two neighbors in the same direction in the 3x3 neighborhood around them lies above the low threshold value. In other words, if they are connected to an active edge. J. Canny recommends that the ratio of high to low threshold limit be in the range two or three to one, based on predicted signal-to-noise ratios. The final output of the third phase consists of a single channel 8-bit unsigned image of 0x00 and 0xFF values based on whether they are accepted or rejected during threshold testing.

Currently only the NPP_BORDER_REPLICATE border type operation is supported. Borderless output can be accomplished by using a larger source image than the destination and adjusting oSrcSize and oSrcOffset parameters accordingly.

Functions

NppStatus **nppiFilterCannyBorderGetBufferSize**(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for the FilterCannyBorder function based on destination image SizeROI width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiFilterCannyBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiDifferentialKernel* eFilterType, *NppiMaskSize* eMaskSize, *Npp16s* nLowThreshold, *Npp16s* nHighThreshold, *NppiNorm* eNorm, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned grayscale to 1 channel 8-bit unsigned black (0x00) and white (0xFF) image with border control.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSrcSize** – Source image width and height in pixels relative to pSrc.
- ▶ **oSrcOffset** – The pixel offset that pSrc points to relative to the origin of the source image.
- ▶ **pDst** – output edge destination_image_pointer.
- ▶ **nDstStep** – output edge destination_image_line_step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eFilterType** – selects between Sobel or Scharr filter type.
- ▶ **eMaskSize** – fixed filter mask size to use.
- ▶ **nLowThreshold** – low hysteresis threshold value.
- ▶ **nHighThreshold** – high hysteresis threshold value.
- ▶ **eNorm** – gradient distance method to use.
- ▶ **eBorderType** – source image border type to use use.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBuffer-Size (see *nppiFilterCannyBorderGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFilterCannyBorder_8u_C3C1R_Ctx( const Npp8u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp8u
*pDst, int nDstStep, NppiSize oSizeROI,
NppiDifferentialKernel eFilterType,
NppiMaskSize eMaskSize, Npp16s
nLowThreshold, Npp16s nHighThreshold,
NppiNorm eNorm, NppiBorderType
eBorderType, Npp8u *pDeviceBuffer,
NppStreamContext nppStreamCtx )
```

3 channel 8-bit unsigned color to 1 channel 8-bit unsigned black (0x00) and white (0xFF) image with border control.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSrcSize** – Source image width and height in pixels relative to pSrc.
- ▶ **oSrcOffset** – The pixel offset that pSrc points to relative to the origin of the source image.

- ▶ **pDst** – output edge destination_image_pointer.
- ▶ **nDstStep** – output edge destination_image_line_step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eFilterType** – selects between Sobel or Scharr filter type.
- ▶ **eMaskSize** – fixed filter mask size to use.
- ▶ **nLowThreshold** – low hysteresis threshold value.
- ▶ **nHighThreshold** – high hysteresis threshold value.
- ▶ **eNorm** – gradient distance method to use.
- ▶ **eBorderType** – source image border type to use use.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBuffer-Size (see *nppiFilterCannyBorderGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.6.4 Image Filter Harris Corners Border

1.10.6.4.1 FilterHarrisCornersBorder

Performs Harris Corner detection on a single channel 8-bit grayscale image and outputs a single channel 32-bit floating point image consisting the corner response at each pixel of the image.

The algorithm consists of two phases. The first phase generates the floating point product of XX, YY, and XY gradients at each pixel in the image. The type of gradient used is controlled by the eFilterType and eMaskSize parameters. The second phase averages those products over a window of either 3x3 or 5x5 pixels around the center pixel then generates the Harris corner response at that pixel which is output in the destination image. The Harris response value is determined as $H = ((XX * YY - XY * XY) - (nK * ((XX + YY) * (XX + YY)))) * nScale$.

Currently only the NPP_BORDER_REPLICATE border type operation is supported. Borderless output can be accomplished by using a larger source image than the destination and adjusting oSrcSize and oSrcOffset parameters accordingly.

Functions

NppStatus **nppiFilterHarrisCornersBorderGetBufferSize**(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for the FilterHarrisCornersBorder function based on destination image SizeROI width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiFilterHarrisCornersBorder_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiDifferentialKernel* eFilterType, *NppiMaskSize* eMaskSize, *NppiMaskSize* eAvgWindowSize, *Npp32f* nK, *Npp32f* nScale, *NppiBorderType* eBorderType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned grayscale to 1 channel 32-bit floating point Harris corners response image with border control.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSrcSize** – Source image width and height in pixels relative to pSrc.
- ▶ **oSrcOffset** – The pixel offset that pSrc points to relative to the origin of the source image.
- ▶ **pDst** – output edge destination_image_pointer.
- ▶ **nDstStep** – output edge destination_image_line_step.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eFilterType** – selects between Sobel or Scharr filter type.
- ▶ **eMaskSize** – fixed filter mask size to use (3x3 or 5x5 for Sobel).
- ▶ **eAvgWindowSize** – fixed window mask size to use (3x3 or 5x5).
- ▶ **nK** – Harris Corners constant (commonly used value is 0.04F).
- ▶ **nScale** – output is scaled by this scale factor.
- ▶ **eBorderType** – source image border type to use.
- ▶ **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBuffer-Size (see *nppiFilterHarrisCornersBorderGetBufferSize()* above)
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

1.10.6.5 Image Filter Hough Line

1.10.6.5.1 FilterHoughLine

Extracts Hough lines from a single channel 8-bit binarized (0, 255) source feature (canny edges, etc.) image.

Outputs a list of lines in point polar format representing the length (rho) and angle (theta) of each line from the origin of the normal to the line using the formula $\rho = x \cos(\theta) + y \sin(\theta)$. The level of discretization, nDelta, is specified as an input parameter. The performance and effectiveness of this function highly depends on this parameter with higher performance for larger numbers and more detailed results for lower numbers. Also, lines are not guaranteed to be added to the pDeviceLines list in the same order from one call to the next. However, all of the same lines will still be generated as long as nMaxLineCount is set large enough so that they all can fit in the list. To convert lines in point polar format back to cartesian lines use the following formula:

```

Npp32f nHough = ((sqrt(2.0F) * static_cast<Npp32f>(oSizeROI.height > oSizeROI.width ?
↳oSizeROI.height
                                                    :
↳oSizeROI.width)) / 2.0F);
int nAccumulatorsHeight = nDelta.rho > 1.0F ? static_cast<int>(ceil(nHough * 2.0F))
                                                    : static_cast<int>(ceil((nHough * 2.0F) /
↳nDelta.rho));
int nCenterX = oSizeROI.width >> 1;
int nCenterY = oSizeROI.height >> 1;
Npp32f nThetaRad = static_cast<Npp32f>(deviceline.theta) * 0.0174532925199433F;
Npp32f nSinTheta = sin(nThetaRad);
Npp32f nCosTheta = cos(nThetaRad);
int nX1, nY1, nX2, nY2;

if (deviceline.theta >= 45 && deviceline.theta <= 135) // degrees
{
    // y = (rho - x cos(theta)) / sin(theta)
    nX1 = minimum cartesian X boundary value;
    nY1 = static_cast<int>((static_cast<Npp32f>(deviceline.rho - (nAccumulatorsHeight
↳>> 1)) -
                                ((nX1 - nCenterX) * nCosTheta)) / nSinTheta + nCenterY);
    nX2 = maximum cartesian X boundary value;
    nY2 = static_cast<int>((static_cast<Npp32f>(deviceline.rho - (nAccumulatorsHeight
↳>> 1)) -
                                ((nX2 - nCenterX) * nCosTheta)) / nSinTheta + nCenterY);
}
else
{
    // x = (rho - y sin(theta)) / cos(theta)
    nY1 = minimum cartesian Y boundary value;
    nX1 = static_cast<int>((static_cast<Npp32f>(deviceline.rho - (nAccumulatorsHeight
↳>> 1)) -
                                ((nY1 - nCenterY) * nSinTheta)) / nCosTheta + nCenterX);
    nY2 = maximum cartesian Y boundary value;
    nX2 = static_cast<int>((static_cast<Npp32f>(deviceline.rho - (nAccumulatorsHeight
↳>> 1)) -
                                ((nY2 - nCenterY) * nSinTheta)) / nCosTheta + nCenterX);
}

```

Functions

NppStatus **nppiFilterHoughLineGetBufferSize**(*NppiSize* oSizeROI, *NppPointPolar* nDelta, int nMaxLineCount, int *hpBufferSize)

Calculate scratch buffer size needed for the FilterHoughLine or FilterHoughLineRegion functions based on destination image SizeROI width and height and nDelta parameters.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nDelta** – rho radial increment and theta angular increment that will be used in the FilterHoughLine or FilterHoughLineRegion function call.
- ▶ **nMaxLineCount** – The maximum number of lines expected from the FilterHoughLine or FilterHoughLineRegion function call.
- ▶ **hpBufferSize** – Required buffer size in bytes. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiFilterHoughLine_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *NppPointPolar* nDelta, int nThreshold, *NppPointPolar* *pDeviceLines, int nMaxLineCount, int *pDeviceLineCount, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned binarized (0, 255) source feature (canny edges, etc.) source image to list of lines in point polar format representing the length (rho) and angle (theta) of each line from the origin of the normal to the line using the formula $\rho = x \cos(\theta) + y \sin(\theta)$.

The level of discretization, nDelta, is specified as an input parameter. The performance and effectiveness of this function highly depends on this parameter with higher performance for larger numbers and more detailed results for lower numbers. nDelta must have the same values as those used in the *nppiFilterHoughLineGetBufferSize()* function call.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nDelta** – Discretization steps, range $0.0F < \text{radial increment } nDelta.\rho < 3.0F$, 1.0F recommended, range $0.25F < \text{angular increment } nDelta.\theta < 3.0F$, 1.0F recommended.
- ▶ **nThreshold** – Minimum number of points to accept a line.
- ▶ **pDeviceLines** – Device pointer to (nMaxLineCount * sizeof(NppPointPolar) line objects.
- ▶ **nMaxLineCount** – The maximum number of lines to output.
- ▶ **pDeviceLineCount** – Device memory pointer to an int that receives the number of lines detected up to nMaxLineCount. Must be allocated with cudaMalloc — passing a host pointer causes NPP_CUDA_KERNEL_EXECUTION_ERROR.

- **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBuffer-Size (see [nppiFilterHoughLineGetBufferSize\(\)](#) above)
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiFilterHoughLineRegion_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *NppPointPolar* nDelta, int nThreshold, *NppPointPolar* *pDeviceLines, *NppPointPolar* oDstROI[2], int nMaxLineCount, int *pDeviceLineCount, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned binarized (0, 255) source feature (canny edges, etc.) source image to list of lines in point polar format representing the length (rho) and angle (theta) of each line from the origin of the normal to the line using the formula $\rho = x \cos(\theta) + y \sin(\theta)$.

The level of discretization, nDelta, is specified as an input parameter. The performance and effectiveness of this function highly depends on this parameter with higher performance for larger numbers and more detailed results for lower numbers. nDelta must have the same values as those used in the [nppiFilterHoughLineGetBufferSize\(\)](#) function call. The oDstROI region limits are used to limit accepted lines to those that fall within those limits.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **nDelta** – Discretization steps, range $0.0F < \text{radial increment } nDelta.\rho < 3.0F$, $1.0F$ recommended, range $0.25F < \text{angular increment } nDelta.\theta < 3.0F$, $1.0F$ recommended.
- **nThreshold** – Minimum number of points to accept a line.
- **pDeviceLines** – Device pointer to (nMaxLineCount * sizeof(NppPointPolar) line objects.
- **oDstROI** – Region limits with $oDstROI[0].\rho \leq \text{accepted } \rho \leq oDstROI[1].\rho$ and $oDstROI[0].\theta \leq \text{accepted } \theta \leq oDstROI[1].\theta$.
- **nMaxLineCount** – The maximum number of lines to output.
- **pDeviceLineCount** – Device memory pointer to an int that receives the number of lines detected up to nMaxLineCount. Must be allocated with cudaMalloc — passing a host pointer causes NPP_CUDA_KERNEL_EXECUTION_ERROR.
- **pDeviceBuffer** – pointer to scratch DEVICE memory buffer of size hpBuffer-Size (see [nppiFilterHoughLineGetBufferSize\(\)](#) above)
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.6.6 Image Filter Histogram Of Oriented Gradients Border

1.10.6.6.1 HistogramOfOrientedGradientsBorder

Performs Histogram Of Oriented Gradients operation on source image generating separate windows of Histogram Descriptors for each requested location.

This function implements the simplest form of functionality described by N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. INRIA, 2005. It supports overlapped contrast normalized block histogram output with L2 normalization only, no threshold clipping, and no pre or post gaussian smoothing of input images or histogram output values. It supports both single channel grayscale source images and three channel color images. For color images, the color channel with the highest magnitude value is used as that pixel's magnitude. Output is row order only. Descriptors are output consecutively with no separation padding if multiple descriptor output is requested (one descriptor per source image location). For example, common HOG parameters are 9 histogram bins per 8 by 8 pixel cell, 2 by 2 cells per block, with a descriptor window size of 64 horizontal by 128 vertical pixels yielding 7 by 15 overlapping blocks (1 cell overlap in both horizontal and vertical directions). This results in 9 bins * 4 cells * 7 horizontal overlapping blocks * 15 vertical overlapping blocks or 3780 32-bit floating point output values (bins) per descriptor window.

The number of horizontal overlapping block histogram bins per descriptor window width is determined by $((\text{oHOGConfig.detectionWindowSize.width} / \text{oHOGConfig.histogramBlockSize}) * 2) - 1) * \text{oHOGConfig.nHistogramBins}$. The number of vertical overlapping block histograms per descriptor window height is determined by $((\text{oHOGConfig.detectionWindowSize.height} / \text{oHOGConfig.histogramBlockSize}) * 2) - 1)$. The offset of each descriptor window in the descriptors output buffer is therefore horizontal histogram bins per descriptor window width * vertical histograms per descriptor window height 32-bit floating point values relative to the previous descriptor window output.

The algorithm uses a 1D centered derivative mask of [-1, 0, +1] when generating input magnitude and angle gradients. Magnitudes are added to the two nearest histogram bins of oriented gradients between 0 and 180 degrees using a weighted linear interpolation of each magnitude value across the 2 nearest angular bin orientations. 2D overlapping blocks of histogram bins consisting of the bins from 2D arrangements of cells are then contrast normalized using L2 normalization and output to the corresponding histogram descriptor window for that particular window location in the window locations list.

Some restrictions include:

```
#define NPP_HOG_MAX_CELL_SIZE                (16)
#define NPP_HOG_MAX_BLOCK_SIZE              (64)
#define NPP_HOG_MAX_BINS_PER_CELL          (16)
#define NPP_HOG_MAX_CELLS_PER_DESCRIPTOR    (256)
#define NPP_HOG_MAX_OVERLAPPING_BLOCKS_PER_DESCRIPTOR (256)
#define NPP_HOG_MAX_DESCRIPTOR_LOCATIONS_PER_CALL (128)
```

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.10.6.6.1.1 Common parameters for `nppiFilterHistogramOfGradientsBorder` functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

The pixel offset that pSrc points to relative to the origin of the source image.

param hpLocations

Host pointer to array of *NppiPoint* source pixel starting locations of requested descriptor windows. Important: hpLocations is a *host pointer*.

param nLocations

Number of *NppiPoint* in pLocations array.

param pDstWindowDescriptorBuffer

Output device memory buffer pointer of size hpDescriptorsSize bytes to first of nLoc descriptor windows (see *nppiHistogramOfGradientsBorderGetDescriptorsSize()* above).

param oSizeROI

Region-Of-Interest (ROI) of source image.

param oHOGConfig

Requested HOG configuration parameters structure.

param pScratchBuffer

Device memory buffer pointer of size hpBufferSize bytes to scratch memory buffer (see *nppiHistogramOfGradientsBorderGetBufferSize()* above).

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus `nppiHistogramOfGradientsBorderGetBufferSize`(const *NppiHOGConfig* oHOGConfig, const *NppiPoint* *hpLocations, int nLocations, *NppiSize* oSizeROI, int *hpBufferSize)

Validates requested HOG configuration and calculates scratch buffer size needed for the HistogramOfGradientsBorder function based on requested HOG configuration, source image ROI, and number and locations of descriptor window locations.

Parameters

- ▶ **oHOGConfig** – Requested HOG configuration parameters structure.
- ▶ **hpLocations** – Host pointer to array of *NppiPoint* source pixel starting locations of requested descriptor windows. Important: hpLocations is a *host pointer*.
- ▶ **nLocations** – Number of *NppiPoint* in pLocations array.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* of source image.
- ▶ **hpBufferSize** – Required buffer size in bytes. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiHistogramOfGradientsBorderGetDescriptorsSize**(const *NppiHOGConfig* oHOGConfig, int nLocations, int *hpDescriptorsSize)

Validates requested HOG configuration and calculates output window descriptors buffer size needed for the HistogramOfGradientsBorder function based on requested HOG configuration, and number of descriptor window locations, one descriptor window is output for each location.

Descriptor windows are located sequentially and contiguously in the descriptors buffer.

The number of horizontal overlapping block histogram bins per descriptor window width is determined by $((\text{oHOGConfig.detectionWindowSize.width} / \text{oHOGConfig.histogramBlockSize}) * 2) - 1$ * oHOGConfig.nHistogramBins. The number of vertical overlapping block histograms per descriptor window height is determined by $((\text{oHOGConfig.detectionWindowSize.height} / \text{oHOGConfig.histogramBlockSize}) * 2) - 1$. The offset of each descriptor window in the descriptors output buffer is therefore horizontal histogram bins per descriptor window width * vertical histograms per descriptor window height floating point values relative to the previous descriptor window output.

Parameters

- ▶ **oHOGConfig** – Requested HOG configuration parameters structure.
- ▶ **nLocations** – Number of *NppiPoint* in pLocations array.
- ▶ **hpDescriptorsSize** – Required buffer size in bytes of output windows descriptors for nLocations descriptor windows. Important: hpDescriptorsSize is a *host pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiHistogramOfGradientsBorder_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, const *NppiPoint* *hpLocations, int nLocations, *Npp32f* *pDstWindowDescriptorBuffer, *NppiSize* oSizeROI, const *NppiHOGConfig* oHOGConfig, *Npp8u* *pScratchBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned grayscale per source image descriptor window location with source image border control to per descriptor window destination floating point histogram of gradients.

Requires first calling `nppiHistogramOfGradientsBorderGetBufferSize` function call to get required scratch (host) working buffer size and `nppiHistogramOfGradientsBorderGetDescriptorsSize()` function call to get total size for nLocations of output histogram block descriptor windows.

For common parameter descriptions, see *Common parameters for nppiFilterHistogramOfGradientsBorder functions*.

```
NppStatus nppiHistogramOfGradientsBorder_8u32f_C3R_Ctx(const Npp8u *pSrc, int
                                                         nSrcStep, NppiSize oSrcSize,
                                                         NppiPoint oSrcOffset, const
                                                         NppiPoint *hpLocations, int
                                                         nLocations, Npp32f
                                                         *pDstWindowDescriptorBuffer,
                                                         NppiSize oSizeROI, const
                                                         NppiHOGConfig oHOGConfig,
                                                         Npp8u *pScratchBuffer,
                                                         NppiBorderType eBorderType,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

3 channel 8-bit unsigned color per source image descriptor window location with source image border control to per descriptor window destination floating point histogram of gradients.

Requires first calling `nppiHistogramOfGradientsBorderGetBufferSize` function call to get required scratch (host) working buffer size and `nppiHistogramOfGradientsBorderGetDescriptorsSize()` function call to get total size for nLocations of output histogram block descriptor windows.

For common parameter descriptions, see *Common parameters for nppiFilterHistogramOfGradientsBorder functions*.

```
NppStatus nppiHistogramOfGradientsBorder_16u32f_C1R_Ctx(const Npp16u *pSrc, int
                                                         nSrcStep, NppiSize oSrcSize,
                                                         NppiPoint oSrcOffset, const
                                                         NppiPoint *hpLocations, int
                                                         nLocations, Npp32f *pDstWin-
                                                         dowDescriptorBuffer,
                                                         NppiSize oSizeROI, const
                                                         NppiHOGConfig oHOGConfig,
                                                         Npp8u *pScratchBuffer,
                                                         NppiBorderType eBorderType,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

1 channel 16-bit unsigned grayscale per source image descriptor window location with source image border control to per descriptor window destination floating point histogram of gradients.

Requires first calling `nppiHistogramOfGradientsBorderGetBufferSize` function call to get required scratch (host) working buffer size and `nppiHistogramOfGradientsBorderGetDescriptorsSize()` function call to get total size for nLocations of output histogram block descriptor windows.

For common parameter descriptions, see *Common parameters for nppiFilterHistogramOfGradientsBorder functions*.

NppStatus **nppiHistogramOfGradientsBorder_16u32f_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, const *NppiPoint* *hpLocations, int nLocations, *Npp32f* *pDstWindowDescriptorBuffer, *NppiSize* oSizeROI, const *NppiHOGConfig* oHOGConfig, *Npp8u* *pScratchBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned color per source image descriptor window location with source image border control to per descriptor window destination floating point histogram of gradients.

Requires first calling `nppiHistogramOfGradientsBorderGetBufferSize` function call to get required scratch (host) working buffer size and *nppiHistogramOfGradientsBorderGetDescriptorsSize()* function call to get total size for nLocations of output histogram block descriptor windows.

For common parameter descriptions, see *Common parameters for nppiFilterHistogramOfGradientsBorder functions*.

NppStatus **nppiHistogramOfGradientsBorder_16s32f_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, const *NppiPoint* *hpLocations, int nLocations, *Npp32f* *pDstWindowDescriptorBuffer, *NppiSize* oSizeROI, const *NppiHOGConfig* oHOGConfig, *Npp8u* *pScratchBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed grayscale per source image descriptor window location with source image border control to per descriptor window destination floating point histogram of gradients.

Requires first calling `nppiHistogramOfGradientsBorderGetBufferSize` function call to get required scratch (host) working buffer size and *nppiHistogramOfGradientsBorderGetDescriptorsSize()* function call to get total size for nLocations of output histogram block descriptor windows.

For common parameter descriptions, see *Common parameters for nppiFilterHistogramOfGradientsBorder functions*.

NppStatus **nppiHistogramOfGradientsBorder_16s32f_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, const *NppiPoint* *hpLocations, int nLocations, *Npp32f* *pDstWindowDescriptorBuffer, *NppiSize* oSizeROI, const *NppiHOGConfig* oHOGConfig, *Npp8u* *pScratchBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed color per source image descriptor window location with source image border control to per descriptor window destination floating point histogram of gradients.

Requires first calling `nppiHistogramOfGradientsBorderGetBufferSize` function call to get required scratch (host) working buffer size and `nppiHistogramOfGradientsBorderGetDescriptorsSize()` function call to get total size for nLocations of output histogram block descriptor windows.

For common parameter descriptions, see *Common parameters for nppiFilterHistogramOfGradientsBorder functions*.

```
NppStatus nppiHistogramOfGradientsBorder_32f_C1R_Ctx(const Npp32f *pSrc, int nSrcStep,
                                                    NppiSize oSrcSize, NppiPoint
                                                    oSrcOffset, const NppiPoint
                                                    *hpLocations, int nLocations,
                                                    Npp32f
                                                    *pDstWindowDescriptorBuffer,
                                                    NppiSize oSizeROI, const
                                                    NppiHOGConfig oHOGConfig,
                                                    Npp8u *pScratchBuffer,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext
                                                    nppStreamCtx)
```

1 channel 32-bit floating point grayscale per source image descriptor window location with source image border control to per descriptor window destination floating point histogram of gradients.

Requires first calling `nppiHistogramOfGradientsBorderGetBufferSize` function call to get required scratch (host) working buffer size and `nppiHistogramOfGradientsBorderGetDescriptorsSize()` function call to get total size for nLocations of output histogram block descriptor windows.

For common parameter descriptions, see *Common parameters for nppiFilterHistogramOfGradientsBorder functions*.

```
NppStatus nppiHistogramOfGradientsBorder_32f_C3R_Ctx(const Npp32f *pSrc, int nSrcStep,
                                                    NppiSize oSrcSize, NppiPoint
                                                    oSrcOffset, const NppiPoint
                                                    *hpLocations, int nLocations,
                                                    Npp32f
                                                    *pDstWindowDescriptorBuffer,
                                                    NppiSize oSizeROI, const
                                                    NppiHOGConfig oHOGConfig,
                                                    Npp8u *pScratchBuffer,
                                                    NppiBorderType eBorderType,
                                                    NppStreamContext
                                                    nppStreamCtx)
```

3 channel 32-bit floating point color per source image descriptor window location with source image border control to per descriptor window destination floating point histogram of gradients.

Requires first calling `nppiHistogramOfGradientsBorderGetBufferSize` function call to get required scratch (host) working buffer size and `nppiHistogramOfGradientsBorderGetDescriptorsSize()` function call to get total size for nLocations of output histogram block descriptor windows.

For common parameter descriptions, see *Common parameters for nppiFilterHistogramOfGradientsBorder functions*.

1.10.7. Image Filter Flood Fill

1.10.7.1 FloodFill

Flood fill a connected region of an image with a specified new value.

FloodFillGetBufferSize

Before calling any of the FloodFill functions the application first needs to call the FloodFillGetBufferSize function to determine the amount of device memory to allocate as a working buffer.

The application allocated device memory is then passed as the pBuffer parameter to the corresponding FloodFill function.

NppStatus **nppiFloodFillGetBufferSize**(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for the FloodFill function based on destination image oSizeROI width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpBufferSize** – Required buffer size in bytes.

1.10.7.2 Flood Fill

1.10.7.2.1 FloodFill

In place flood fill of a region of pixels connected to the pixel at the seed pixel location in an image with a new pixel value.

Before calling any of the FloodFill functions the application first needs to call the FloodFillGetBufferSize function to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding FloodFill function.

Optionally the function can return connected region information for the filled region in the pConnectedRegion *NppiConnectedRegion* structure in host memory. The oBoundingBox x and y will be set to the left and top coordinates and width and height will be set to the right bottom coordinates of the bounding box relative to pSrcDst. Set pConnectedRegion to NULL if not required. Requesting pConnectedRegion information may slightly degrade performance.

Functions

NppStatus **nppiFloodFill_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, const *Npp8u* nNewValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus `nppiFloodFill_8u_C3IR_Ctx`(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, const *Npp8u* aNewValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus `nppiFloodFill_16u_C1IR_Ctx`(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, const *Npp16u* nNewValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFloodFill_16u_C3IR_Ctx(Npp16u *pSrcDst, int nSrcDstStep, NppiPoint oSeed,  
                                     const Npp16u aNewValues[3], NppiNorm eNorm,  
                                     NppiSize oSizeROI, NppiConnectedRegion  
                                     *pConnectedRegion, Npp8u *pBuffer,  
                                     NppStreamContext nppStreamCtx)
```

3 channel 16-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFloodFill_32u_C1IR_Ctx(Npp32u *pSrcDst, int nSrcDstStep, NppiPoint oSeed,
                                     const Npp32u nNewValue, NppiNorm eNorm, NppiSize
                                     oSizeROI, NppiConnectedRegion *pConnectedRegion,
                                     Npp8u *pBuffer, NppStreamContext nppStreamCtx)
```

1 channel 32-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding *FloodFillGetBufferSize* call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFloodFill_32u_C3IR_Ctx(Npp32u *pSrcDst, int nSrcDstStep, NppiPoint oSeed,
                                     const Npp32u aNewValues[3], NppiNorm eNorm,
                                     NppiSize oSizeROI, NppiConnectedRegion
                                     *pConnectedRegion, Npp8u *pBuffer,
                                     NppStreamContext nppStreamCtx)
```

3 channel 32-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding *FloodFillGetBufferSize* call.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.7.3 Flood Fill Boundary

1.10.7.3.1 FloodFillBoundary

In place flood fill of a region of pixels connected to the pixel at the seed pixel location in an image with a new pixel value. Also fill boundary of filled region with a specified color.

Before calling any of the FloodFill functions the application first needs to call the FloodFillGetBufferSize function to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding FloodFill function.

Optionally the function can return connected region information for the filled region in the pConnectedRegion *NppiConnectedRegion* structure in host memory. The oBoundingBox x and y will be set to the left and top coordinates and width and height will be set to the right bottom coordinates of the bounding box relative to pSrcDst. Set pConnectedRegion to NULL if not required. Requesting pConnectedRegion information may slightly degrade performance.

Functions

NppStatus **nppiFloodFillBoundary_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, const *Npp8u* nNewValue, const *Npp8u* nBoundaryValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer grayscale in place flood fill.

Parameters

- **pSrcDst** – *In-Place Image Pointer.*
- **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- **oSeed** – Image location of seed pixel value to be used for comparison.
- **nNewValue** – Image pixel value to be used to replace matching pixels.
- **nBoundaryValue** – Image pixel value to be used for region boundary.
- **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- **oSizeROI** – *Region-Of-Interest (ROI).*
- **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiFloodFillBoundary_8u_C3IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, const *Npp8u* aNewValues[3], const *Npp8u* aBoundaryValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **aBoundaryValues** – Image pixel values to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding *FloodFillGetBufferSize* call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiFloodFillBoundary_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, const *Npp16u* nNewValue, const *Npp16u* nBoundaryValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **nBoundaryValue** – Image pixel value to be used for region boundary.

- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus `nppiFloodFillBoundary_16u_C3IR_Ctx`(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, const *Npp16u* aNewValues[3], const *Npp16u* aBoundaryValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **aBoundaryValues** – Image pixel values to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus `nppiFloodFillBoundary_32u_C1IR_Ctx`(*Npp32u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, const *Npp32u* nNewValue, const *Npp32u* nBoundaryValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 32-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **nBoundaryValue** – Image pixel value to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus `nppiFloodFillBoundary_32u_C3IR_Ctx`(*Npp32u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, const *Npp32u* aNewValues[3], const *Npp32u* aBoundaryValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 32-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **aBoundaryValues** – Image pixel values to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.

- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.7.4 Flood Fill Range

1.10.7.4.1 FloodFillRange

In place flood fill of a region of pixels connected to the pixel at the seed pixel location and with a Pixel value $\geq$ Min and $\leq$ Max in an image with a new pixel value.

Before calling any of the FloodFill functions the application first needs to call the FloodFillGetBufferSize function to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding FloodFill function.

Optionally the function can return connected region information for the filled region in the pConnectedRegion *NppiConnectedRegion* structure in host memory. The oBoundingBox x and y will be set to the left and top coordinates and width and height will be set to the right bottom coordinates of the bounding box relative to pSrcDst. Set pConnectedRegion to NULL if not required. Requesting pConnectedRegion information may slightly degrade performance.

Functions

NppStatus **nppiFloodFillRange_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp8u* nMin, *Npp8u* nMax, const *Npp8u* nNewValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be $\geq$ this value.
- ▶ **nMax** – Value of tested pixel must be $\leq$ this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.

- **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiFloodFillRange_8u_C3IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp8u* aMin[3], *Npp8u* aMax[3], const *Npp8u* aNewValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer color in place flood fill.

Parameters

- **pSrcDst** – *In-Place Image Pointer*.
- **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- **oSeed** – Image location of seed pixel value to be used for comparison.
- **aMin** – Value of each element of tested pixel must be >= the corresponding aMin value.
- **aMax** – Value of each element of tested pixel must be <= the corresponding aMax value.
- **aNewValues** – Image pixel values to be used to replace matching pixels.
- **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- **oSizeROI** – *Region-Of-Interest (ROI)*.
- **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiFloodFillRange_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp16u* nMin, *Npp16u* nMax, const *Npp16u* nNewValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer grayscale in place flood fill.

Parameters

- **pSrcDst** – *In-Place Image Pointer*.

- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be $\geq$ this value.
- ▶ **nMax** – Value of tested pixel must be $\leq$ this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiFloodFillRange_16u_C3IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp16u* aMin[3], *Npp16u* aMax[3], const *Npp16u* aNewValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be $\geq$ the corresponding aMin value.
- ▶ **aMax** – Value of each element of tested pixel must be $\leq$ the corresponding aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiFloodFillRange_32u_C1IR_Ctx**(*Npp32u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp32u* nMin, *Npp32u* nMax, const *Npp32u* nNewValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 32-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be >= this value.
- ▶ **nMax** – Value of tested pixel must be <= this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding *FloodFillGetBufferSize* call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiFloodFillRange_32u_C3IR_Ctx**(*Npp32u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp32u* aMin[3], *Npp32u* aMax[3], const *Npp32u* aNewValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 32-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be >= the corresponding aMin value.

- ▶ **aMax** – Value of each element of tested pixel must be $\leq$ the corresponding aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.7.5 Flood Fill Range Boundary

1.10.7.5.1 FloodFillRangeBoundary

In place flood fill of a region of pixels connected to the pixel at the seed pixel location and with a Pixel value $\geq$ Min and $\leq$ Max in an image with a new pixel value. Also fill boundary of filled region with a specified color.

Before calling any of the FloodFill functions the application first needs to call the `FloodFillGetBufferSize` function to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the `pBuffer` parameter to the corresponding FloodFill function.

Optionally the function can return connected region information for the filled region in the `pConnectedRegion` *NppiConnectedRegion* structure in host memory. The `oBoundingBox` `x` and `y` will be set to the left and top coordinates and `width` and `height` will be set to the right bottom coordinates of the bounding box relative to `pSrcDst`. Set `pConnectedRegion` to NULL if not required. Requesting `pConnectedRegion` information may slightly degrade performance.

Functions

```
NppStatus nppiFloodFillRangeBoundary_8u_C1IR_Ctx(Npp8u *pSrcDst, int nSrcDstStep,  
                                                NppiPoint oSeed, Npp8u nMin, Npp8u  
                                                nMax, const Npp8u nNewValue, const  
                                                Npp8u nBoundaryValue, NppiNorm  
                                                eNorm, NppiSize oSizeROI,  
                                                NppiConnectedRegion  
                                                *pConnectedRegion, Npp8u *pBuffer,  
                                                NppStreamContext nppStreamCtx)
```

1 channel 8-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.

- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be $\geq$ this value.
- ▶ **nMax** – Value of tested pixel must be $\leq$ this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **nBoundaryValue** – Image pixel value to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus `nppiFloodFillRangeBoundary_8u_C3IR_Ctx`(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp8u* aMin[3], *Npp8u* aMax[3], const *Npp8u* aNewValues[3], const *Npp8u* aBoundaryValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be $\geq$ the corresponding aMin value.
- ▶ **aMax** – Value of each element of tested pixel must be $\leq$ the corresponding aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **aBoundaryValues** – Image pixel values to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiFloodFillRangeBoundary_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp16u* nMin, *Npp16u* nMax, const *Npp16u* nNewValue, const *Npp16u* nBoundaryValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be >= this value.
- ▶ **nMax** – Value of tested pixel must be <= this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **nBoundaryValue** – Image pixel value to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFloodFillRangeBoundary_16u_C3IR_Ctx(Npp16u *pSrcDst, int nSrcDstStep,
NppiPoint oSeed, Npp16u aMin[3],
Npp16u aMax[3], const Npp16u
aNewValues[3], const Npp16u
aBoundaryValues[3], NppiNorm
eNorm, NppiSize oSizeROI,
NppiConnectedRegion
*pConnectedRegion, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

3 channel 16-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be >= the corresponding aMin value.
- ▶ **aMax** – Value of each element of tested pixel must be <= the corresponding aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **aBoundaryValues** – Image pixel values to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFloodFillRangeBoundary_32u_C1IR_Ctx(Npp32u *pSrcDst, int nSrcDstStep,
NppiPoint oSeed, Npp32u nMin,
Npp32u nMax, const Npp32u
nNewValue, const Npp32u
nBoundaryValue, NppiNorm eNorm,
NppiSize oSizeROI,
NppiConnectedRegion
*pConnectedRegion, Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

1 channel 32-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.

- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be $\geq$ this value.
- ▶ **nMax** – Value of tested pixel must be $\leq$ this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **nBoundaryValue** – Image pixel value to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus `nppiFloodFillRangeBoundary_32u_C3IR_Ctx`(*Npp32u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp32u* aMin[3], *Npp32u* aMax[3], const *Npp32u* aNewValues[3], const *Npp32u* aBoundaryValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 32-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be $\geq$ the corresponding aMin value.
- ▶ **aMax** – Value of each element of tested pixel must be $\leq$ the corresponding aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **aBoundaryValues** – Image pixel values to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.7.6 Flood Fill Gradient**1.10.7.6.1 FloodFillGradient**

In place flood fill of a region of pixels connected to the pixel at the seed pixel location and with an original pixel value including seed - Min to seed + Max in an image with a new pixel value.

In place flood fill of a region of pixels connected to the pixel at the seed pixel location in an image with a new pixel value.

Before calling any of the FloodFill functions the application first needs to call the FloodFillGetBufferSize function to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding FloodFill function.

Optionally the function can return connected region information for the filled region in the pConnectedRegion *NppiConnectedRegion* structure in host memory. The oBoundingBox x and y will be set to the left and top coordinates and width and height will be set to the right bottom coordinates of the bounding box relative to pSrcDst. Set pConnectedRegion to NULL if not required. Requesting pConnectedRegion information may slightly degrade performance.

Functions

NppStatus **nppiFloodFillGradient_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp8u* nMin, *Npp8u* nMax, const *Npp8u* nNewValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be $\geq$ seed value - this value.
- ▶ **nMax** – Value of tested pixel must be $\leq$ seed value + this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.

- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus `nppiFloodFillGradient_8u_C3IR_Ctx`(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp8u* aMin[3], *Npp8u* aMax[3], const *Npp8u* aNewValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be $\geq$ the corresponding seed value - aMin value.
- ▶ **aMax** – Value of each element of tested pixel must be $\leq$ the corresponding seed value + aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFloodFillGradient_16u_C1IR_Ctx(Npp16u *pSrcDst, int nSrcDstStep,
                                             NppiPoint oSeed, Npp16u nMin, Npp16u
                                             nMax, const Npp16u nNewValue, NppiNorm
                                             eNorm, NppiSize oSizeROI,
                                             NppiConnectedRegion *pConnectedRegion,
                                             Npp8u *pBuffer, NppStreamContext
                                             nppStreamCtx)
```

1 channel 16-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be $\geq$ seed value - this value.
- ▶ **nMax** – Value of tested pixel must be $\leq$ seed value + this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding *FloodFillGetBufferSize* call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFloodFillGradient_16u_C3IR_Ctx(Npp16u *pSrcDst, int nSrcDstStep,
                                             NppiPoint oSeed, Npp16u aMin[3], Npp16u
                                             aMax[3], const Npp16u aNewValues[3],
                                             NppiNorm eNorm, NppiSize oSizeROI,
                                             NppiConnectedRegion *pConnectedRegion,
                                             Npp8u *pBuffer, NppStreamContext
                                             nppStreamCtx)
```

3 channel 16-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be $\geq$ the corresponding seed value - aMin value.

- ▶ **aMax** – Value of each element of tested pixel must be $\leq$ the corresponding seed value + aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus `nppiFloodFillGradient_32u_C1IR_Ctx`(*Npp32u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp32u* nMin, *Npp32u* nMax, const *Npp32u* nNewValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 32-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be $\geq$ seed value - this value.
- ▶ **nMax** – Value of tested pixel must be $\leq$ seed value + this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes


```
NppStatus nppiFloodFillGradient_32u_C3IR_Ctx(Npp32u *pSrcDst, int nSrcDstStep,
NppiPoint oSeed, Npp32u aMin[3], Npp32u
aMax[3], const Npp32u aNewValues[3],
NppiNorm eNorm, NppiSize oSizeROI,
NppiConnectedRegion *pConnectedRegion,
Npp8u *pBuffer, NppStreamContext
nppStreamCtx)
```

3 channel 32-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be $\geq$ the corresponding seed value - aMin value.
- ▶ **aMax** – Value of each element of tested pixel must be $\leq$ the corresponding seed value + aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.7.7 Flood Fill Gradient Boundary

1.10.7.7.1 FloodFillGradientBoundary

In place flood fill of a region of pixels connected to the pixel at the seed pixel location and with an original pixel value including seed - Min to seed + Max in an image with a new pixel value. Also fill boundary of filled region with a specified color.

In place flood fill of a region of pixels connected to the pixel at the seed pixel location in an image with a new pixel value.

Before calling any of the FloodFill functions the application first needs to call the FloodFillGetBufferSize function to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding FloodFill function.

Optionally the function can return connected region information for the filled region in the pConnectedRegion *NppiConnectedRegion* structure in host memory. The oBoundingBox x and y will be set to

the left and top coordinates and width and height will be set to the right bottom coordinates of the bounding box relative to pSrcDst. Set pConnectedRegion to NULL if not required. Requesting pConnectedRegion information may slightly degrade performance.

Functions

NppStatus **nppiFloodFillGradientBoundary_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp8u* nMin, *Npp8u* nMax, const *Npp8u* nNewValue, const *Npp8u* nBoundaryValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be $\geq$ seed value - this value.
- ▶ **nMax** – Value of tested pixel must be $\leq$ seed value + this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **nBoundaryValue** – Image pixel value to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFloodFillGradientBoundary_8u_C3IR_Ctx(Npp8u *pSrcDst, int nSrcDstStep,
NppiPoint oSeed, Npp8u aMin[3],
Npp8u aMax[3], const Npp8u
aNewValues[3], const Npp8u
aBoundaryValues[3], NppiNorm
eNorm, NppiSize oSizeROI,
NppiConnectedRegion
*pConnectedRegion, Npp8u
*pBuffer, NppStreamContext
nppStreamCtx)
```

3 channel 8-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be $\geq$ the corresponding seed value - aMin value.
- ▶ **aMax** – Value of each element of tested pixel must be $\leq$ the corresponding seed value + aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **aBoundaryValues** – Image pixel values to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFloodFillGradientBoundary_16u_C1IR_Ctx(Npp16u *pSrcDst, int
nSrcDstStep, NppiPoint oSeed,
Npp16u nMin, Npp16u nMax,
const Npp16u nNewValue, const
Npp16u nBoundaryValue,
NppiNorm eNorm, NppiSize
oSizeROI, NppiConnectedRegion
*pConnectedRegion, Npp8u
*pBuffer, NppStreamContext
nppStreamCtx)
```

1 channel 16-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be $\geq$ seed value - this value.
- ▶ **nMax** – Value of tested pixel must be $\leq$ seed value + this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **nBoundaryValue** – Image pixel value to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, `nppiNormInf` will use 8 way connectivity and `nppiNormL1` will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding `FloodFillGetBufferSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus `nppiFloodFillGradientBoundary_16u_C3IR_Ctx`(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp16u* aMin[3], *Npp16u* aMax[3], const *Npp16u* aNewValues[3], const *Npp16u* aBoundaryValues[3], *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be $\geq$ the corresponding seed value - aMin value.
- ▶ **aMax** – Value of each element of tested pixel must be $\leq$ the corresponding seed value + aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **aBoundaryValues** – Image pixel values to be used for region boundary.

- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus nppiFloodFillGradientBoundary_32u_C1IR_Ctx(*Npp32u* *pSrcDst, int nSrcDstStep, *NppiPoint* oSeed, *Npp32u* nMin, *Npp32u* nMax, const *Npp32u* nNewValue, const *Npp32u* nBoundaryValue, *NppiNorm* eNorm, *NppiSize* oSizeROI, *NppiConnectedRegion* *pConnectedRegion, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 32-bit unsigned integer grayscale in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **nMin** – Value of tested pixel must be $\geq$ seed value - this value.
- ▶ **nMax** – Value of tested pixel must be $\leq$ seed value + this value.
- ▶ **nNewValue** – Image pixel value to be used to replace matching pixels.
- ▶ **nBoundaryValue** – Image pixel value to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiFloodFillGradientBoundary_32u_C3IR_Ctx(Npp32u *pSrcDst, int  
nSrcDstStep, NppiPoint oSeed,  
Npp32u aMin[3], Npp32u aMax[3],  
const Npp32u aNewValues[3],  
const Npp32u  
aBoundaryValues[3], NppiNorm  
eNorm, NppiSize oSizeROI,  
NppiConnectedRegion  
*pConnectedRegion, Npp8u  
*pBuffer, NppStreamContext  
nppStreamCtx)
```

3 channel 32-bit unsigned integer color in place flood fill.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *In-Place-Image Line Step* in bytes.
- ▶ **oSeed** – Image location of seed pixel value to be used for comparison.
- ▶ **aMin** – Value of each element of tested pixel must be $\geq$ the corresponding seed value - aMin value.
- ▶ **aMax** – Value of each element of tested pixel must be $\leq$ the corresponding seed value + aMax value.
- ▶ **aNewValues** – Image pixel values to be used to replace matching pixels.
- ▶ **aBoundaryValues** – Image pixel values to be used for region boundary.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pConnectedRegion** – Optional host memory pointer to an *NppiConnectedRegion* object which returns information about the filled region. Set to NULL if not needed.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding FloodFillGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.8. Label Markers

1.10.8.1 LabelMarkers

Generate image connected region label markers to be used for later image segmentation.

These functions have been deprecated. Use LabelMarkersUF functions instead.

LabelMarkersUFGetBufferSize

Before calling any of the LabelMarkersUF functions the application first needs to call the LabelMarkersUFGetBufferSize function to determine the amount of device memory to allocate as a working buffer.

The application allocated device memory is then passed as the pBuffer parameter to the corresponding LabelMarkersUF function.

NppStatus **nppiLabelMarkersUFGetBufferSize_32u_C1R**(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed 1 channel 32-bit unsigned integer LabelMarkersUF function based on destination image oSizeROI width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpBufferSize** – Required buffer size in bytes.

1.10.8.2 Label MarkersUF

1.10.8.2.1 LabelMarkersUF

Generate image connected region label markers to be used for later image segmentation.

A connected region is any pixel region where all pixels in the region have the same pixel value. Note that marker label IDs generally increase in value from image left to right and top to bottom they are not generated in any particular order and there may be numeric gaps between sequential marker IDs. To limit the number of marker IDs generated the application should pass the image through a threshold filter before calling this function. Doing so however does not necessarily limit the maximum marker ID value generated by this function. Note that these functions currently only support image ROI sizes up to 4 gigapixels. Also note that while these functions support destination image pitches that are not exactly equal to oSizeROI.width * sizeof(Npp32u) it can avoid a cudaMemcpyAsync() call to copy the final result from the working buffer to the destination image if the pitch exactly equals oSizeROI.width * sizeof(Npp32u).

Before calling any of the LabelMarkersUF functions the application first needs to call the LabelMarkersUFGetBufferSize to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding LabelMarkersUF function.

The algorithm used in this implementation is based on the one described in “An Optimized Union-Find Algorithm for Connected Components Labeling Using GPUs” by Jun Chen and others.

Note that the destination image in these functions must be allocated with cudaMalloc() and NOT cudaMallocPitch(). Also the pitch of the output image MUST be set to oSizeROI.width * sizeof(Npp32u).

Functions

NppStatus **nppiLabelMarkersUF_8u32u_C1R_Ctx**(*Npp8u* *pSrc, int nSrcStep, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiNorm* eNorm, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit to 32-bit unsigned integer label markers image generation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding *LabelMarkersUFGetBufferSize* call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiLabelMarkersUF_16u32u_C1R_Ctx**(*Npp16u* *pSrc, int nSrcStep, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiNorm* eNorm, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit to 32-bit unsigned integer label markers image generation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding *LabelMarkersUFGetBufferSize* call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiLabelMarkersUF_32u_C1R_Ctx**(*Npp32u* *pSrc, int nSrcStep, *Npp32u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiNorm* eNorm, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 32-bit to 32-bit unsigned integer label markers image generation.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **eNorm** – Type of pixel connectivity test to use, *nppiNormInf* will use 8 way connectivity and *nppiNormL1* will use 4 way connectivity.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding *LabelMarkersUFGetBufferSize* call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.8.3 Label MarkersUF Batch

1.10.8.3.1 LabelMarkersUFBatch

Generate image connected region label markers to be used for later image segmentation for a batched list of images.

A connected region is any pixel region where all pixels in the region have the same pixel value. Note that marker label IDs generally increase in value from image left to right and top to bottom they are not generated in any particular order and there may be numeric gaps between sequential marker IDs. To limit the number of marker IDs generated the application should pass the image through a threshold filter before calling this function. Doing so however does not necessarily limit the maximum marker ID value generated by this function. Note that these functions currently only support image ROI sizes up to 4 gigapixels. Also note that these functions work directly in the destination image so destination pitch MUST be equal to destination ROI.width * sizeof(Npp32u) and even if destination pitch is greater than that the output will be in that format. If this doesn't work for you use the single image version of these functions or *cudaMemcpy2D* can be called for each output image to restore the original pitch. Note that all source and destination images in the batch list must contain enough device memory to support the full ROI size. ROIs in descriptor lists are ignored in these functions.

The algorithm used in this implementation is based on the one described in “An Optimized Union-Find Algorithm for Connected Components Labeling Using GPUs” by Jun Chen and others.

Note that the destination images in these functions must be allocated with *cudaMalloc()* and NOT *cudaMallocPitch()*. Also the pitch of the output image MUST be set to *oSizeROI.width * sizeof(Npp32u)*.

Functions

NppStatus **nppiLabelMarkersUFBatch_8u32u_C1R_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppiNorm* eNorm, *NppStreamContext* nppStreamCtx)

1 channel 8-bit to 32-bit unsigned integer label markers image generation with fixed destination ROI applied to all images in the batch.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory source image descriptors, oSize element is ignored.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer* device memory pointer to the list of device memory destination image descriptors, oSize element is ignored.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity for all images in batch.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiLabelMarkersUFBatch_16u32u_C1R_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppiNorm* eNorm, *NppStreamContext* nppStreamCtx)

1 channel 16-bit to 32-bit unsigned integer label markers image generation with fixed destination ROI applied to all images in the batch.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory source image descriptors, oSize element is ignored.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer* device memory pointer to the list of device memory destination image descriptors, oSize element is ignored.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity for all images in batch.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiLabelMarkersUFBatch_32u_C1R_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oSizeROI, *NppiNorm* eNorm, *NppStreamContext* nppStreamCtx)

1 channel 32-bit to 32-bit unsigned integer label markers image generation with fixed destination ROI applied to all images in the batch.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory source image descriptors, oSize element is ignored.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer* device memory pointer to the list of device memory destination image descriptors, oSize element is ignored.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity for all images in batch.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Image Data Related Error Codes, ROI Related Error Codes***1.10.8.4 Label MarkersUF Batch Advanced****1.10.8.4.1 LabelMarkersUFBatchAdvanced**

Generate image connected region label markers to be used for later image segmentation for a batched list of images.

A connected region is any pixel region where all pixels in the region have the same pixel value. Note that marker label IDs generally increase in value from image left to right and top to bottom they are not generated in any particular order and there may be numeric gaps between sequential marker IDs. To limit the number of marker IDs generated the application should pass the image through a threshold filter before calling this function. Doing so however does not necessarily limit the maximum marker ID value generated by this function. Note that these functions currently only support image ROI sizes up to 4 gigapixels. Also note that these functions work directly in the destination image so destination pitch MUST be equal to destination ROI.width * sizeof(Npp32u) for each destination image in the list and even if the destination image pitch is greater than that of the output will be in that format. If this doesn't work for you use the single image version of these functions or cudaMemcpy2D can be called for each output image to restore the original pitch. Note that all source and destination images in the batch list must contain enough device memory to contain their specified ROI size.

The algorithm used in this implementation is based on the one described in “An Optimized Union-Find Algorithm for Connected Components Labeling Using GPUs” by Jun Chen and others.

Note that the destination images in these functions must be allocated with cudaMalloc() and NOT cudaMallocPitch(). Also the pitch of the output image MUST be set to oSizeROI.width * sizeof(Npp32u).

Functions

NppStatus **nppiLabelMarkersUFBatch_8u32u_C1R_Advanced_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppiNorm* eNorm, *NppStreamContext* nppStreamCtx)

1 channel 8-bit to 32-bit unsigned integer label markers image generation with per image destination ROI.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory source image descriptors, oSize element is ignored.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer* device memory pointer to the list of device memory destination image descriptors.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)* maximum ROI width and height of ALL images in the batch.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity for all images in batch.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiLabelMarkersUFBatch_16u32u_C1R_Advanced_Ctx**(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppiNorm* eNorm, *NppStreamContext* nppStreamCtx)

1 channel 16-bit to 32-bit unsigned integer label markers image generation with per image destination ROI.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory source image descriptors, oSize element is ignored.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer* device memory pointer to the list of device memory destination image descriptors.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).

- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)* maximum ROI width and height of ALL images in the batch.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity for all images in batch.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus nppiLabelMarkersUFBatch_32u_C1R_Advanced_Ctx(const *NppImageDescriptor* *pSrcBatchList, *NppImageDescriptor* *pDstBatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *NppiNorm* eNorm, *NppStreamContext* nppStreamCtx)

1 channel 32-bit to 32-bit unsigned integer label markers image generation with per image destination ROI.

Parameters

- ▶ **pSrcBatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory source image descriptors, oSize element is ignored.
- ▶ **pDstBatchList** – *Destination-Batch-Images Pointer* device memory pointer to the list of device memory destination image descriptors.
- ▶ **nBatchSize** – Number of *NppImageDescriptor* structures processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)* maximum ROI width and height of ALL images in the batch.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity for all images in batch.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.10.8.5 Image Filter Compress Marker Labels

Removes sparseness between marker label IDs output from LabelMarkers call.

CompressMarkerLabelsGetBufferSize

Before calling any of the CompressMarkerLabels functions the application first needs to call the corresponding CompressMarkerLabelsGetBufferSize function to determine the amount of device memory to allocate as a working buffer.

The application allocated device memory is then passed as the pBuffer parameter to the corresponding CompressMarkerLabels function.

NppStatus **nppiCompressMarkerLabelsGetBufferSize_32u_C1R**(int nStartingNumber, int *hpBufferSize)

Calculate scratch buffer size needed for 1 channel 32-bit unsigned integer CompressMarkerLabels function based on the number returned in pNumber from a previous nppiLabelMarkers call.

Note that this is the only function that supports the nppiCompressMarkerLabelsUF_32u function and that nStartingNumber MUST be ROI width * ROI height when used with that function.

Parameters

- ▶ **nStartingNumber** – The value returned from a previous call to the nppiLabelMarkers_32u function or ROI width * ROI height for images generated by the nppiLabelMarkersUF function and those values MUST match the values used those used in that call.
- ▶ **hpBufferSize** – Required buffer size in bytes.

CompressMarkerLabels

NOTE: The previous versions of these functions have been deprecated.

Only nppiCompressMarkerLabelsUF_32u works with output from nppiLabelMarkersUF functions.

NppStatus **nppiCompressMarkerLabelsUF_32u_C1IR_Ctx**(*Npp32u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, int nStartingNumber, int *pNewNumber, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

1 channel 32-bit unsigned integer in place connected region marker label renumbering for output from nppiLabelMarkersUF functions only with numbering sparseness elimination.

Note that the image in this function must be allocated with cudaMalloc() and NOT cudaMallocPitch(). Also the pitch MUST be set to oSizeROI.width * sizeof(Npp32u). And the image pointer and oSizeROI values MUST match those used when nppiLabelMarkersUF was called.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *Source-Image Line Step*. NOTE THAT THIS VALUE MUST BE EQUAL TO oSizeROI.width * sizeof(Npp32u).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.

- ▶ **nStartingNumber** – MUST be ROI width * ROI height and MUST match ROI values used in the label markers generation function call.
- ▶ **pNewNumber** – Pointer to host memory integer value where the maximum renumbered marker label ID will be returned.
- ▶ **pBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding CompressMarkerLabelsGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.10.8.6 Image Filter Compressed Marker Labels Info

Various methods for extracting information from compressed marker labels.

1.10.8.7 Image Filter Contour Pixel Interpolation

1.10.8.7.1 ContourPixelInterpolation

Various functions for interpolating pixels in image contours.

1.10.8.8 Contours Image Marching Squares Interpolation

1.10.8.8.1 ContoursImageMarchingSquaresInterpolation

Apply Marching Squares bilinear interpolation to all contour pixels in a contours image where contour pixels have values of 0 and non-contour pixels have values of 255.

Functions

NppStatus nppiContoursImageMarchingSquaresInterpolation_32f_C1R_Ctx(*Npp8u*
 *pContoursIm-
 ageDev,
Npp32s
 nContoursIm-
 ageStep,
NppiPoint32f
 *pContoursIn-
 terpolatedIm-
 ageDev,
Npp32s
 nContoursIn-
 terpolatedIm-
 ageStep,
*NppiCon-
 tourPixelDirec-
 tionInfo*
 *pContoursDi-
 rectionIm-
 ageDev,
Npp32s
 nContoursDi-
 rectionIm-
 ageStep,
*NppiCon-
 tourPixelGe-
 ometryInfo*
 *pContour-
 sPixelGeome-
 tryListsDev,
*NppiCon-
 tourPixelGe-
 ometryInfo*
 *pContour-
 sPixelGeome-
 tryListsHost,
NppiPoint32f
 *pContoursIn-
 terpolatedGe-
 ometryLists-
 Dev, *Npp32u*
 *pContour-
 sPixels-
 FoundListHost,
Npp32u
 *pContour-
 sPixelsStartin-
 gOffsetDev,
Npp32u
 *pContour-
 sPixelsStartin-
 gOffsetHost,
Npp32u nTo-
 tallImagePixel-
 ContourCount,
Npp32u
 nMaxLabel,
 LabelID,
Npp32u nFirst-
 ContourGeom-

1 channel integer coordinate based contours image to marching squares bilinear interpolated coordinates contours image.

Note that ALL input and output data for the function MUST be in device memory except where noted otherwise. Also nFirstContourID and nLastContourID allow only a portion of the contour geometry lists in the image to be output.

Note that the geometry list for each contour will begin at pContoursGeometryList-sHost[pContoursPixelStartingOffsetHost[nContourID] * sizeof(NppiContourPixelGeometryInfo).

Note that to significantly improve performance by default a contour that contains more than 256K pixels will be bypassed when generating the output geometry list. The contour ID and number of contour pixels will be output in the contour list however. You can still get this function to output the geometry of this size contour however by calling the function with a starting contour ID of that contour ID and ending contour ID of that contour ID + 1. Note that doing so for contours approaching a million pixels can take many minutes. Also, due to the structure of some images contour ID 0 can contain ALL contours in the image so setting the starting contour ID to 1 can significantly increase output preprocessing performance.

Also note that the ordered contour geometry list is contained in the oContourOrderedGeometry-Location object within the *NppiContourPixelGeometryInfo* object for each contour pixel and that this location information is the only valid information in the object relevant to that ordered pixel.

Note that for a particular contour geometry pixel list the bounding box for that contour will be contained in the following elements of first pixel (pixel 0) of the corresponding contour pixel list.

```
pixel[0].oContourPrevPixelLocation.x will contain pCurMarkerLabelsInfo->
↳oMarkerLabelBoundingBox.x      upper left x
pixel[0].oContourPrevPixelLocation.y will contain pCurMarkerLabelsInfo->
↳oMarkerLabelBoundingBox.y      upper left y
pixel[0].oContourNextPixelLocation.x will contain pCurMarkerLabelsInfo->
↳oMarkerLabelBoundingBox.width  lower right x
pixel[0].oContourNextPixelLocation.y will contain pCurMarkerLabelsInfo->
↳oMarkerLabelBoundingBox.height lower right y
```

Note that while the bounding box is relatively accurate occasionally a few contour pixels may extend beyond the bounding box limits. Additionally, bounding box coordinates may vary slightly between runs on the same input due to the parallel reduction used internally; pixel counts (nMarkerLabelPixelCount) remain deterministic. If exact reproducible bounding boxes are required, derive them from the contour pixel geometry lists instead.

Parameters

- ▶ **pContoursImageDev** – pointer to device memory image of at least oSizeROI.width * sizeof(Npp8u) * oSizeROI.height bytes.
- ▶ **nContoursImageStep** – image line step.
- ▶ **pContoursInterpolatedImageDev** – pointer to device memory image of at least oSizeROI.width * sizeof(NppiPoint32f) * OSizeROI.height bytes.
- ▶ **nContoursInterpolatedImageStep** – image line step
- ▶ **pContoursDirectionImageDev** – *Source-Image Pointer* to output image in device memory containing per contour pixel direction info around each uniquely labeled connected pixel region returned by corresponding nppiCompressedMarkerLabelsUFIInfo call.

- ▶ **nContoursDirectionImageStep** – image_line_step for contours image.
- ▶ **pContoursPixelGeometryListsDev** – pointer to device memory buffer allocated to be at least as big as size returned by corresponding nppiCompressedMarkerLabelsUFGGetGeometryListsSize call.
- ▶ **pContoursPixelGeometryListsHost** – pointer to host memory buffer allocated to be at least as big as size returned by corresponding nppiCompressedMarkerLabelsUFGGetGeometryListsSize call.
- ▶ **pContoursInterpolatedGeometryListsDev** – pointer to device memory buffer allocated to be at least nTotalImagePixelContourCount * sizeof(NppPoint32f) bytes.
- ▶ **pContoursPixelsFoundListHost** – host memory pointer to array of nMaxMarkerLabelID unsigned integers returned by previous call to nppiCompressedMarkerLabelsUFGContoursPixelGeometryLists_C1R_Ctx.
- ▶ **pContoursPixelsStartingOffsetDev** – device memory pointer to array of unsigned integers returned by this call representing the starting offset index of each contour found during geometry list generation.
- ▶ **pContoursPixelsStartingOffsetHost** – host memory pointer to array of unsigned integers returned by this call representing the starting offset index of each contour found during geometry list generation.
- ▶ **nTotalImagePixelContourCount** – the total number of contour pixels in the image returned by nppiCompressedMarkerLabelsUFGInfo_32u_C1R_Ctx() call.
- ▶ **nMaxMarkerLabelID** – the value of the maximum marker label ID returned by corresponding compress marker labels UF call.
- ▶ **nFirstContourGeometryListID** – the ID of the first contour geometry list to output.
- ▶ **nLastContourGeometryListID** – the ID of the last contour geometry list to output, last ID MUST be greater than first ID.
- ▶ **pContoursBlockSegmentListDev** – device memory pointer to array of *NppiContourBlockSegment* objects, contents will be initialized by NPP.
- ▶ **pContoursBlockSegmentListHost** – host memory pointer to array of *NppiContourBlockSegment* objects, contents will be initialized by NPP.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* for the images, must be the same as used in previous calls.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus nppiContoursImageMarchingSquaresInterpolation_64f_C1R_Ctx(*Npp8u*
 *pContoursIm-
 ageDev,
Npp32s
 nContoursIm-
 ageStep,
NppiPoint64f
 *pContoursIn-
 terpolatedIm-
 ageDev,
Npp32s
 nContoursIn-
 terpolatedIm-
 ageStep,
*NppiCon-
 tourPixelDirec-
 tionInfo*
 *pContoursDi-
 rectionIm-
 ageDev,
Npp32s
 nContoursDi-
 rectionIm-
 ageStep,
*NppiCon-
 tourPixelGe-
 ometryInfo*
 *pContour-
 sPixelGeome-
 tryListsDev,
*NppiCon-
 tourPixelGe-
 ometryInfo*
 *pContour-
 sPixelGeome-
 tryListsHost,
NppiPoint64f
 *pContoursIn-
 terpolatedGe-
 ometryLists-
 Dev, *Npp32u*
 *pContour-
 sPixels-
 FoundListHost,
Npp32u
 *pContour-
 sPixelsStartin-
 gOffsetDev,
Npp32u
 *pContour-
 sPixelsStartin-
 gOffsetHost,
Npp32u nTo-
 tallImagePixel-
 ContourCount,
Npp32u
 nMaxMarker
 LabelID,
Npp32u nFirst-
 ContourGeom-

Note that while the bounding box is relatively accurate occasionally a few contour pixels may extend beyond the bounding box limits.

Additionally, bounding box coordinates may vary slightly between runs on the same input due to the parallel reduction used internally; pixel counts (nMarkerLabelPixelCount) remain deterministic. If exact reproducible bounding boxes are required, derive them from the contour pixel geometry lists instead.

Parameters

- ▶ **pContoursImageDev** – pointer to device memory image of at least `oSizeROI.width * sizeof(Npp8u) * oSizeROI.height` bytes.
- ▶ **nContoursImageStep** – image line step.
- ▶ **pContoursInterpolatedImageDev** – pointer to device memory image of at least `oSizeROI.width * sizeof(NppiPoint64f) * OSizeROI.height` bytes.
- ▶ **nContoursInterpolatedImageStep** – image line step
- ▶ **pContoursDirectionImageDev** – *Source-Image Pointer* to output image in device memory containing per contour pixel direction info around each uniquely labeled connected pixel region returned by corresponding `nppiCompressedMarkerLabelsUInfo` call.
- ▶ **nContoursDirectionImageStep** – `image_line_step` for contours image.
- ▶ **pContoursPixelGeometryListsDev** – pointer to device memory buffer allocated to be at least as big as size returned by corresponding `nppiCompressedMarkerLabelsUFGGetGeometryListsSize` call.
- ▶ **pContoursPixelGeometryListsHost** – pointer to host memory buffer allocated to be at least as big as size returned by corresponding `nppiCompressedMarkerLabelsUFGGetGeometryListsSize` call.
- ▶ **pContoursInterpolatedGeometryListsDev** – pointer to device memory buffer allocated to be at least `nTotalImagePixelContourCount * sizeof(NppPoint64f)` bytes.
- ▶ **pContoursPixelsFoundListHost** – host memory pointer to array of `nMaxMarkerLabelID` unsigned integers returned by previous call to `nppiCompressedMarkerLabelsUFGContoursPixelGeometryLists_C1R_Ctx`.
- ▶ **pContoursPixelsStartingOffsetDev** – device memory pointer to array of unsigned integers returned by this call representing the starting offset index of each contour found during geometry list generation.
- ▶ **pContoursPixelsStartingOffsetHost** – host memory pointer to array of unsigned integers returned by this call representing the starting offset index of each contour found during geometry list generation.
- ▶ **nTotalImagePixelContourCount** – the total number of contour pixels in the image returned by `nppiCompressedMarkerLabelsUInfo_32u_C1R_Ctx()` call.
- ▶ **nMaxMarkerLabelID** – the value of the maximum marker label ID returned by corresponding compress marker labels UF call.
- ▶ **nFirstContourGeometryListID** – the ID of the first contour geometry list to output.

- ▶ **nLastContourGeometryListID** – the ID of the last contour geometry list to output, last ID MUST be greater than first ID.
- ▶ **pContoursBlockSegmentListDev** – device memory pointer to array of *Nppi-ContourBlockSegment* objects, contents will be initialized by NPP.
- ▶ **pContoursBlockSegmentListHost** – host memory pointer to array of *Nppi-ContourBlockSegment* objects, contents will be initialized by NPP.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* for the images, must be the same as used in previous calls.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.10.9. Bound Segments

Note that these functions have been deprecated.

Use the `nppiLabelMarkerUF`, `nppiCompressMarkerLabelsUF`, and `nppiCompressedMarkerLabelsUFInfo` functions to generate connected pixel region boundaries (contours).

1.10.10. Watershed Segmentation

1.10.10.1 WatershedSegmentation

Segments a grayscale image using the watershed segmentation technique described in “Efficient 2D and 3D Watershed on Graphics Processing Unit: Block-Asynchronous Approaches Based on Cellular Automata” by Pablo Quesada-Barriuso and others.

SegmentWatershedGetBufferSize

Before calling any of the `SegmentWatershed` functions the application first needs to call the corresponding `SegmentWatershedGetBufferSize` function to determine the amount of device memory to allocate as a working buffer.

The application allocated device memory is then passed as the `pBuffer` parameter to the corresponding `SegmentWatershed` function.

NppStatus **nppiSegmentWatershedGetBufferSize_8u_C1R**(*NppiSize* oSizeROI, size_t *hpDeviceMemoryBufferSize)

Calculate scratch buffer sizes needed for 1 channel 8-bit unsigned integer watershed segmentation function based on destination image `oSizeROI` width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpDeviceMemoryBufferSize** – Required device memory buffer size in bytes.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSegmentWatershedGetBufferSize_16u_C1R**(*NppiSize* oSizeROI, size_t *hpDeviceMemoryBufferSize)

Calculate scratch buffer sizes needed for 1 channel 16-bit unsigned integer watershed segmentation function based on destination image oSizeROI width and height.

Parameters

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpDeviceMemoryBufferSize** – Required device memory buffer size in bytes.

Returns

Image Data Related Error Codes, ROI Related Error Codes

SegmentWatershed

Generate an output image containing regions of constant value grayscale defined by watershed segmentation plateau boundaries from a grayscale input image.

Optionally output the corresponding marker labels image.

Before calling any of the SegmentWatershed functions the application first needs to call the corresponding SegmentWatershedGetBufferSize to determine the amount of device memory to allocate as working buffers. The allocated memory is then passed as the pDeviceMemoryBuffer parameter to the corresponding SegmentWatershed function.

NppStatus **nppiSegmentWatershed_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, *Npp32s* nSrcDstStep, *Npp32u* *pMarkerLabels, *Npp32s* nMarkerLabelsStep, *NppiNorm* eNorm, *NppiWatershedSegmentBoundaryType* eSegmentBoundaryType, *NppiSize* oSizeROI, *Npp8u* *pDeviceMemoryBuffer, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer in place image watershed segmentation generation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer*.
- ▶ **nSrcDstStep** – *Source-Image Line Step*.
- ▶ **pMarkerLabels** – Device memory pointer to optionally output the corresponding marker labels image, set to NULL if no marker labels image output is desired.
- ▶ **nMarkerLabelsStep** – Marker labels image line step, ignored if pMarkerLabels is NULL.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity.
- ▶ **eSegmentBoundaryType** – Type of segment boundaries, if any, to be added to output image.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* for both segmented image and corresponding marker labels image.

- ▶ **pDeviceMemoryBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding SegmentWatershedGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

NppStatus **nppiSegmentWatershed_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, *Npp32s* nSrcDstStep, *Npp32u* *pMarkerLabels, *Npp32s* nMarkerLabelsStep, *NppiNorm* eNorm, *NppiWatershedSegmentBoundaryType* eSegmentBoundaryType, *NppiSize* oSizeROI, *Npp8u* *pDeviceMemoryBuffer, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer in place image watershed segmentation generation.

Parameters

- ▶ **pSrcDst** – *In-Place Image Pointer.*
- ▶ **nSrcDstStep** – *Source-Image Line Step.*
- ▶ **pMarkerLabels** – Device memory pointer to optionally output the corresponding marker labels image, set to NULL if no marker labels image output is desired.
- ▶ **nMarkerLabelsStep** – Marker labels image line step, ignored if pMarkerLabels is NULL.
- ▶ **eNorm** – Type of pixel connectivity test to use, nppiNormInf will use 8 way connectivity and nppiNormL1 will use 4 way connectivity.
- ▶ **eSegmentBoundaryType** – Type of segment boundaries, if any, to be added to output image.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* for both segmented image and corresponding marker labels image.
- ▶ **pDeviceMemoryBuffer** – Pointer to device memory scratch buffer at least as large as value returned by the corresponding SegmentWatershedGetBufferSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

1.11. Image Geometry Transforms Functions

Routines manipulating an image geometry.

These functions can be found in the nppig library. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

1.11.1. Geometric Transform API Specifics

This section covers some of the unique API features common to the geometric transform primitives.

1.11.2. Geometric Transforms and ROIs

Geometric transforms operate on source and destination ROIs. The way these ROIs affect the processing of pixels differs from other (non geometric) image-processing primitives: Only pixels in the intersection of the destination ROI and the transformed source ROI are being processed.

The typical processing proceeds as follows:

1. Transform the rectangular source ROI (given in source image coordinates) into the destination image space. This yields a quadrilateral.
2. Write only pixels in the intersection of the transformed source ROI and the destination ROI.

1.11.3. Pixel Interpolation

The majority of image geometry transform operation need to perform a resampling of the source image as source and destination pixels are not coincident.

NPP supports the following pixel interpolation modes (in order from fastest to slowest and lowest to highest quality):

- ▶ nearest neighbor
- ▶ linear interpolation
- ▶ cubic convolution
- ▶ supersampling
- ▶ interpolation using Lanczos window function

1.11.4. Resize Error Codes

The resize primitives return the following error codes:

- NPP_WRONG_INTERSECTION_ROI_ERROR indicates an error condition if srcROIrect has no intersection with the source image.
- NPP_RESIZE_NO_OPERATION_ERROR if either destination ROI width or height is less than 1 pixel.
- NPP_RESIZE_FACTOR_ERROR Indicates an error condition if either nXFactor or nYFactor is less than or equal to zero or in the case of NPPI_INTER_SUPER are not both downscaling.
- NPP_INTERPOLATION_ERROR if eInterpolation has an illegal value.
- NPP_SIZE_ERROR if source size width or height is less than 2 pixels.
- NPP_RECTANGLE_ERROR if either destination ROI width or height is less than 1 pixel.

1.11.5. ResizeSqrPixel

ResizeSqrPixel functions attempt to choose source pixels that would approximately represent the center of the destination pixels. It does so by using the following scaling formula to select source pixels for interpolation:

```
nAdjustedXFactor = 1.0 / nXFactor;
nAdjustedYFactor = 1.0 / nYFactor;
nAdjustedXShift = nXShift * nAdjustedXFactor + ((1.0 - nAdjustedXFactor) * 0.5);
nAdjustedYShift = nYShift * nAdjustedYFactor + ((1.0 - nAdjustedYFactor) * 0.5);
nSrcX = nAdjustedXFactor * nDstX - nAdjustedXShift;
nSrcY = nAdjustedYFactor * nDstY - nAdjustedYShift;
```

ResizeSqrPixel functions support the following interpolation modes:

```
NPPI_INTER_NN
NPPI_INTER_LINEAR
NPPI_INTER_CUBIC
NPPI_INTER_CUBIC2P_BSPLINE
NPPI_INTER_CUBIC2P_CATMULLROM
NPPI_INTER_CUBIC2P_B05C03
NPPI_INTER_SUPER
NPPI_INTER_LANCZOS
```

In the ResizeSqrPixel functions below source image clip checking is handled as follows:

If the source pixel fractional x and y coordinates are greater than or equal to oSizeROI.x and less than oSizeROI.x + oSizeROI.width and greater than or equal to oSizeROI.y and less than oSizeROI.y + oSizeROI.height then the source pixel is considered to be within the source image clip rectangle and the source image is sampled. Otherwise the source image is not sampled and a destination pixel is not written to the destination image.

GetResizeRect

Returns *NppiRect* which represents the offset and size of the destination rectangle that would be generated by resizing the source *NppiRect* by the requested scale factors and shifts.

NppStatus **nppiGetResizeRect**(*NppiRect* oSrcROI, *NppiRect* *pDstRect, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)

Parameters

- ▶ **oSrcROI** – Region of interest in the source image.
- ▶ **pDstRect** – User supplied host memory pointer to an *NppiRect* structure that will be filled in by this function with the region of interest in the destination image.
- ▶ **nXFactor** – Factor by which x dimension is changed.
- ▶ **nYFactor** – Factor by which y dimension is changed.
- ▶ **nXShift** – Source pixel shift in x-direction.
- ▶ **nYShift** – Source pixel shift in y-direction.
- ▶ **eInterpolation** – The type of eInterpolation to perform resampling.

Returns

Image Data Related Error Codes, ROI Related Error Codes, Resize Error Codes

ResizeSqrPixel

Resizes images.

1.11.6. Common parameters for nppiResizeSqrPackedPixel functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Size in pixels of the source image.

param oSrcROI

Region of interest in the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstROI

Region of interest in the destination image.

param nXFactor

Factor by which x dimension is changed.

param nYFactor

Factor by which y dimension is changed.

param nXShift

Source pixel shift in x-direction.

param nYShift

Source pixel shift in y-direction.

param eInterpolation

The type of eInterpolation to perform resampling.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Resize Error Codes

1.11.7. Common parameters for nppiResizeSqrPlanarPixel functions:

param pSrc

Source-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param nSrcStep

Source-Image Line Step.

param oSrcSize

Size in pixels of the source image.

param oSrcROI

Region of interest in the source image.

param pDst

Destination-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param nDstStep

Destination-Image Line Step.

param oDstROI

Region of interest in the destination image.

param nXFactor

Factor by which x dimension is changed.

param nYFactor

Factor by which y dimension is changed.

param nXShift

Source pixel shift in x-direction.

param nYShift

Source pixel shift in y-direction.

param eInterpolation

The type of eInterpolation to perform resampling.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Resize Error Codes

NppStatus **nppiResizeSqrPixel_8u_C1R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions:*.

NppStatus **nppiResizeSqrPixel_8u_C3R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions:*.

NppStatus **nppiResizeSqrPixel_8u_C4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions:*.

NppStatus **nppiResizeSqrPixel_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image resize not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions:*.

NppStatus **nppiResizeSqrPixel_8u_P3R_Ctx**(const *Npp8u* *const pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst[3], int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar image resize.

For common parameter descriptions, see [Common parameters for nppiResizeSqrPlanarPixel functions](#).

```
NppStatus nppiResizeSqrPixel_8u_P4R_Ctx(const Npp8u *const pSrc[4], NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp8u *pDst[4], int
                                         nDstStep, NppiRect oDstROI, double nXFactor,
                                         double nYFactor, double nXShift, double nYShift,
                                         int eInterpolation, NppStreamContext
                                         nppStreamCtx)
```

4 channel 8-bit unsigned planar image resize.

For common parameter descriptions, see [Common parameters for nppiResizeSqrPlanarPixel functions](#).

```
NppStatus nppiResizeSqrPixel_16u_C1R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int
                                           nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int
                                           nDstStep, NppiRect oDstROI, double nXFactor,
                                           double nYFactor, double nXShift, double nYShift,
                                           int eInterpolation, NppStreamContext
                                           nppStreamCtx)
```

1 channel 16-bit unsigned image resize.

For common parameter descriptions, see [Common parameters for nppiResizeSqrPackedPixel functions](#).

```
NppStatus nppiResizeSqrPixel_16u_C3R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int
                                           nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int
                                           nDstStep, NppiRect oDstROI, double nXFactor,
                                           double nYFactor, double nXShift, double nYShift,
                                           int eInterpolation, NppStreamContext
                                           nppStreamCtx)
```

3 channel 16-bit unsigned image resize.

For common parameter descriptions, see [Common parameters for nppiResizeSqrPackedPixel functions](#).

```
NppStatus nppiResizeSqrPixel_16u_C4R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int
                                           nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int
                                           nDstStep, NppiRect oDstROI, double nXFactor,
                                           double nYFactor, double nXShift, double nYShift,
                                           int eInterpolation, NppStreamContext
                                           nppStreamCtx)
```

4 channel 16-bit unsigned image resize.

For common parameter descriptions, see [Common parameters for nppiResizeSqrPackedPixel functions](#).

```
NppStatus nppiResizeSqrPixel_16u_AC4R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int
                                           nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int
                                           nDstStep, NppiRect oDstROI, double nXFactor,
                                           double nYFactor, double nXShift, double nYShift,
                                           int eInterpolation, NppStreamContext
                                           nppStreamCtx)
```

4 channel 16-bit unsigned image resize not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*:

```
NppStatus nppiResizeSqrPixel_16u_P3R_Ctx(const Npp16u *const pSrc[3], NppiSize oSrcSize,
                                         int nSrcStep, NppiRect oSrcROI, Npp16u *pDst[3],
                                         int nDstStep, NppiRect oDstROI, double nXFactor,
                                         double nYFactor, double nXShift, double nYShift,
                                         int eInterpolation, NppStreamContext
                                         nppStreamCtx)
```

3 channel 16-bit unsigned planar image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPlanarPixel functions*:

```
NppStatus nppiResizeSqrPixel_16u_P4R_Ctx(const Npp16u *const pSrc[4], NppiSize oSrcSize,
                                         int nSrcStep, NppiRect oSrcROI, Npp16u *pDst[4],
                                         int nDstStep, NppiRect oDstROI, double nXFactor,
                                         double nYFactor, double nXShift, double nYShift,
                                         int eInterpolation, NppStreamContext
                                         nppStreamCtx)
```

4 channel 16-bit unsigned planar image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPlanarPixel functions*:

```
NppStatus nppiResizeSqrPixel_16s_C1R_Ctx(const Npp16s *pSrc, NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp16s *pDst, int
                                         nDstStep, NppiRect oDstROI, double nXFactor,
                                         double nYFactor, double nXShift, double nYShift,
                                         int eInterpolation, NppStreamContext
                                         nppStreamCtx)
```

1 channel 16-bit signed image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*:

```
NppStatus nppiResizeSqrPixel_16s_C3R_Ctx(const Npp16s *pSrc, NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp16s *pDst, int
                                         nDstStep, NppiRect oDstROI, double nXFactor,
                                         double nYFactor, double nXShift, double nYShift,
                                         int eInterpolation, NppStreamContext
                                         nppStreamCtx)
```

3 channel 16-bit signed image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*:

```
NppStatus nppiResizeSqrPixel_16s_C4R_Ctx(const Npp16s *pSrc, NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp16s *pDst, int
                                         nDstStep, NppiRect oDstROI, double nXFactor,
                                         double nYFactor, double nXShift, double nYShift,
                                         int eInterpolation, NppStreamContext
                                         nppStreamCtx)
```

4 channel 16-bit signed image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*:

NppStatus nppiResizeSqrPixel_16s_AC4R_Ctx(const *Npp16s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16s* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed image resize not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*.

NppStatus nppiResizeSqrPixel_16s_P3R_Ctx(const *Npp16s* *const pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16s* *pDst[3], int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed planar image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPlanarPixel functions*.

NppStatus nppiResizeSqrPixel_16s_P4R_Ctx(const *Npp16s* *const pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16s* *pDst[4], int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed planar image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPlanarPixel functions*.

NppStatus nppiResizeSqrPixel_32f_C1R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*.

NppStatus nppiResizeSqrPixel_32f_C3R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*.

NppStatus **nppiResizeSqrPixel_32f_C4R_Ctx**(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*.

NppStatus **nppiResizeSqrPixel_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image resize not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*.

NppStatus **nppiResizeSqrPixel_32f_P3R_Ctx**(const *Npp32f* *const pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst[3], int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point planar image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPlanarPixel functions*.

NppStatus **nppiResizeSqrPixel_32f_P4R_Ctx**(const *Npp32f* *const pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst[4], int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point planar image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPlanarPixel functions*.

NppStatus **nppiResizeSqrPixel_64f_C1R_Ctx**(const *Npp64f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp64f* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 64-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*.

NppStatus nppiResizeSqrPixel_64f_C3R_Ctx(const *Npp64f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp64f* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 64-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*.

NppStatus nppiResizeSqrPixel_64f_C4R_Ctx(const *Npp64f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp64f* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 64-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*.

NppStatus nppiResizeSqrPixel_64f_AC4R_Ctx(const *Npp64f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp64f* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 64-bit floating point image resize not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPackedPixel functions*.

NppStatus nppiResizeSqrPixel_64f_P3R_Ctx(const *Npp64f* *const pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp64f* *pDst[3], int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 64-bit floating point planar image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPlanarPixel functions*.

NppStatus nppiResizeSqrPixel_64f_P4R_Ctx(const *Npp64f* *const pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp64f* *pDst[4], int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 64-bit floating point planar image resize.

For common parameter descriptions, see *Common parameters for nppiResizeSqrPlanarPixel functions*.

NppStatus nppiResizeAdvancedGetBufferHostSize_8u_C1R(*NppiSize* oSrcROI, *NppiSize* oDstROI, int *hpBufferSize, int eInterpolationMode)

Buffer size for *nppiResizeSqrPixel_8u_C1R_Advanced_Ctx*.

Parameters

- ▶ **oSrcROI** – *Region-Of-Interest (ROI)*.
- ▶ **oDstROI** – *Region-Of-Interest (ROI)*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **eInterpolationMode** – The type of eInterpolation to perform resampling. Currently only supports NPPI_INTER_LANCZOS3_Advanced.

Returns

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes*.

NppStatus **nppiResizeSqrPixel_8u_C1R_Advanced_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, double nXFactor, double nYFactor, *Npp8u* *pBuffer, int eInterpolationMode, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned image resize.

This primitive matches the behavior of GraphicsMagick++.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **oSrcSize** – Size in pixels of the source image.
- ▶ **oSrcROI** – Region of interest in the source image.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oDstROI** – Region of interest in the destination image.
- ▶ **nXFactor** – Factor by which x dimension is changed.
- ▶ **nYFactor** – Factor by which y dimension is changed.
- ▶ **pBuffer** – Device buffer that is used during calculations.
- ▶ **eInterpolationMode** – The type of eInterpolation to perform resampling. Currently only supports NPPI_INTER_LANCZOS3_Advanced.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes, Resize Error Codes

1.11.8. Resize

Resize functions use scale factor automatically determined by the width and height ratios of input and output *Region-Of-Interest (ROI)*.

This simplified function replaces the previous version which was deprecated in an earlier release. In this function the resize scale factor is automatically determined by the width and height ratios of oSrcRectROI and oDstRectROI. If either of those parameters intersect their respective image sizes then pixels outside the image size width and height will not be processed.

Resize supports the following interpolation modes:

```
NPPI_INTER_NN
NPPI_INTER_LINEAR
NPPI_INTER_CUBIC
NPPI_INTER_SUPER
NPPI_INTER_LANCZOS
```

The resize primitives return the following error codes: *Resize Error Codes*.

GetResizeTiledSourceOffset

Helper function that can be used when tiling a destination image with a source image using multiple Resize calls.

oSrcRectROI and oDstRectROI widths and heights should remain unmodified even if they will overlap source and destination image sizes. oDstRectROI offsets should be set to the destination offset of the new tile. Resize function processing will stop when source or destination image sizes are reached, any unavailable source image pixels beyond source image size will be border replicated. There is no particular association assumed between source and destination image locations. The values of oSrcRectROI.x and oSrcRectROI.y are ignored during this function call.

NppStatus nppiGetResizeTiledSourceOffset(*NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, *NppiPoint* *pNewSrcRectOffset)

Parameters

- ▶ **oSrcRectROI** – Region of interest in the source image (may overlap source image size width and height).
- ▶ **oDstRectROI** – Region of interest in the destination image (may overlap destination image size width and height).
- ▶ **pNewSrcRectOffset** – Pointer to host memory *NppiPoint* object that will contain the new source image ROI offset to be used in the nppiResize call to generate that tile.

Returns

Image Data Related Error Codes, ROI Related Error Codes, Resize Error Codes

Resize

Resizes images.

1.11.9. Common parameters for nppiResize packed pixel functions:

param pSrc

Source-Image Pointer to origin of source image.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Size in pixels of the entire source image.

param oSrcRectROI

Region of interest in the source image (may overlap source image size width and height).

param pDst

Destination-Image Pointer to origin of destination image.

param nDstStep

Destination-Image Line Step.

param oDstSize

Size in pixels of the entire destination image.

param oDstRectROI

Region of interest in the destination image (may overlap destination image size width and height).

param eInterpolation

The type of eInterpolation to perform resampling (16f versions do not support Lanczos interpolation).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Resize Error Codes

1.11.10. Common parameters for nppiResize planar pixel functions:

param pSrc

Source-Planar-Image Pointer Array (host memory array containing device memory image plane origin pointers).

param nSrcStep*Source-Image Line Step.***param oSrcSize**

Size in pixels of the entire source image.

param oSrcRectROI

Region of interest in the source image (may overlap source image size width and height).

param pDst*Destination-Planar-Image Pointer Array* (host memory array containing device memory image plane origin pointers).**param nDstStep***Destination-Image Line Step.***param oDstSize**

Size in pixels of the entire destination image.

param oDstRectROI

Region of interest in the destination image (may overlap destination image size width and height).

param eInterpolation

The type of eInterpolation to perform resampling.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes, Resize Error Codes*

NppStatus **nppiResize_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiRect* oSrcRectROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions:*.

NppStatus **nppiResize_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiRect* oSrcRectROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions:*.

NppStatus **nppiResize_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiRect* oSrcRectROI, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions:*.

```
NppStatus nppiResize_8u_AC4R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp8u *pDst, int nDstStep, NppiSize  
                                oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

4 channel 8-bit unsigned image resize not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_8u_P3R_Ctx(const Npp8u *pSrc[3], int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp8u *pDst[3], int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

3 channel 8-bit unsigned planar image resize.

For common parameter descriptions, see *Common parameters for nppiResize planar pixel functions*.

```
NppStatus nppiResize_8u_P4R_Ctx(const Npp8u *pSrc[4], int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp8u *pDst[4], int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

4 channel 8-bit unsigned planar image resize.

For common parameter descriptions, see *Common parameters for nppiResize planar pixel functions*.

```
NppStatus nppiResize_16u_C1R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp16u *pDst, int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

1 channel 16-bit unsigned image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_16u_C3R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp16u *pDst, int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

3 channel 16-bit unsigned image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_16u_C4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp16u *pDst, int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

4 channel 16-bit unsigned image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_16u_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp16u *pDst, int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int  
                                eInterpolation, NppStreamContext nppStreamCtx)
```

4 channel 16-bit unsigned image resize not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_16u_P3R_Ctx(const Npp16u *pSrc[3], int nSrcStep, NppiSize oSrcSize,
                                   NppiRect oSrcRectROI, Npp16u *pDst[3], int nDstStep,
                                   NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,
                                   NppStreamContext nppStreamCtx)
```

3 channel 16-bit unsigned planar image resize.

For common parameter descriptions, see *Common parameters for nppiResize planar pixel functions*.

```
NppStatus nppiResize_16u_P4R_Ctx(const Npp16u *pSrc[4], int nSrcStep, NppiSize oSrcSize,
                                   NppiRect oSrcRectROI, Npp16u *pDst[4], int nDstStep,
                                   NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,
                                   NppStreamContext nppStreamCtx)
```

4 channel 16-bit unsigned planar image resize.

For common parameter descriptions, see *Common parameters for nppiResize planar pixel functions*.

```
NppStatus nppiResize_16s_C1R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize,
                                   NppiRect oSrcRectROI, Npp16s *pDst, int nDstStep,
                                   NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,
                                   NppStreamContext nppStreamCtx)
```

1 channel 16-bit signed image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_16s_C3R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize,
                                   NppiRect oSrcRectROI, Npp16s *pDst, int nDstStep,
                                   NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,
                                   NppStreamContext nppStreamCtx)
```

3 channel 16-bit signed image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_16s_C4R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize,
                                   NppiRect oSrcRectROI, Npp16s *pDst, int nDstStep,
                                   NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,
                                   NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_16s_AC4R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize oSrcSize,
                                   NppiRect oSrcRectROI, Npp16s *pDst, int nDstStep,
                                   NppiSize oDstSize, NppiRect oDstRectROI, int
                                   eInterpolation, NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed image resize not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.


```
NppStatus nppiResize_16s_P3R_Ctx(const Npp16s *pSrc[3], int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp16s *pDst[3], int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

3 channel 16-bit signed planar image resize.

For common parameter descriptions, see *Common parameters for nppiResize planar pixel functions*.

```
NppStatus nppiResize_16s_P4R_Ctx(const Npp16s *pSrc[4], int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp16s *pDst[4], int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed planar image resize.

For common parameter descriptions, see *Common parameters for nppiResize planar pixel functions*.

```
NppStatus nppiResize_16f_C1R_Ctx(const Npp16f *pSrc, int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp16f *pDst, int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

1 channel 16-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_16f_C3R_Ctx(const Npp16f *pSrc, int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp16f *pDst, int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

3 channel 16-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_16f_C4R_Ctx(const Npp16f *pSrc, int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp16f *pDst, int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

4 channel 16-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_32f_C1R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp32f *pDst, int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

1 channel 32-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions*.

```
NppStatus nppiResize_32f_C3R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize,  
                                NppiRect oSrcRectROI, Npp32f *pDst, int nDstStep,  
                                NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```


3 channel 32-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions:*.

```
NppStatus nppiResize_32f_C4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize,
                                   NppiRect oSrcRectROI, Npp32f *pDst, int nDstStep,
                                   NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,
                                   NppStreamContext nppStreamCtx)
```

4 channel 32-bit floating point image resize.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions:*.

```
NppStatus nppiResize_32f_AC4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize,
                                   NppiRect oSrcRectROI, Npp32f *pDst, int nDstStep,
                                   NppiSize oDstSize, NppiRect oDstRectROI, int
                                   eInterpolation, NppStreamContext nppStreamCtx)
```

4 channel 32-bit floating point image resize not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResize packed pixel functions:*.

```
NppStatus nppiResize_32f_P3R_Ctx(const Npp32f *pSrc[3], int nSrcStep, NppiSize oSrcSize,
                                   NppiRect oSrcRectROI, Npp32f *pDst[3], int nDstStep,
                                   NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,
                                   NppStreamContext nppStreamCtx)
```

3 channel 32-bit floating point planar image resize.

For common parameter descriptions, see *Common parameters for nppiResize planar pixel functions:*.

```
NppStatus nppiResize_32f_P4R_Ctx(const Npp32f *pSrc[4], int nSrcStep, NppiSize oSrcSize,
                                   NppiRect oSrcRectROI, Npp32f *pDst[4], int nDstStep,
                                   NppiSize oDstSize, NppiRect oDstRectROI, int eInterpolation,
                                   NppStreamContext nppStreamCtx)
```

4 channel 32-bit floating point planar image resize.

For common parameter descriptions, see *Common parameters for nppiResize planar pixel functions:*.

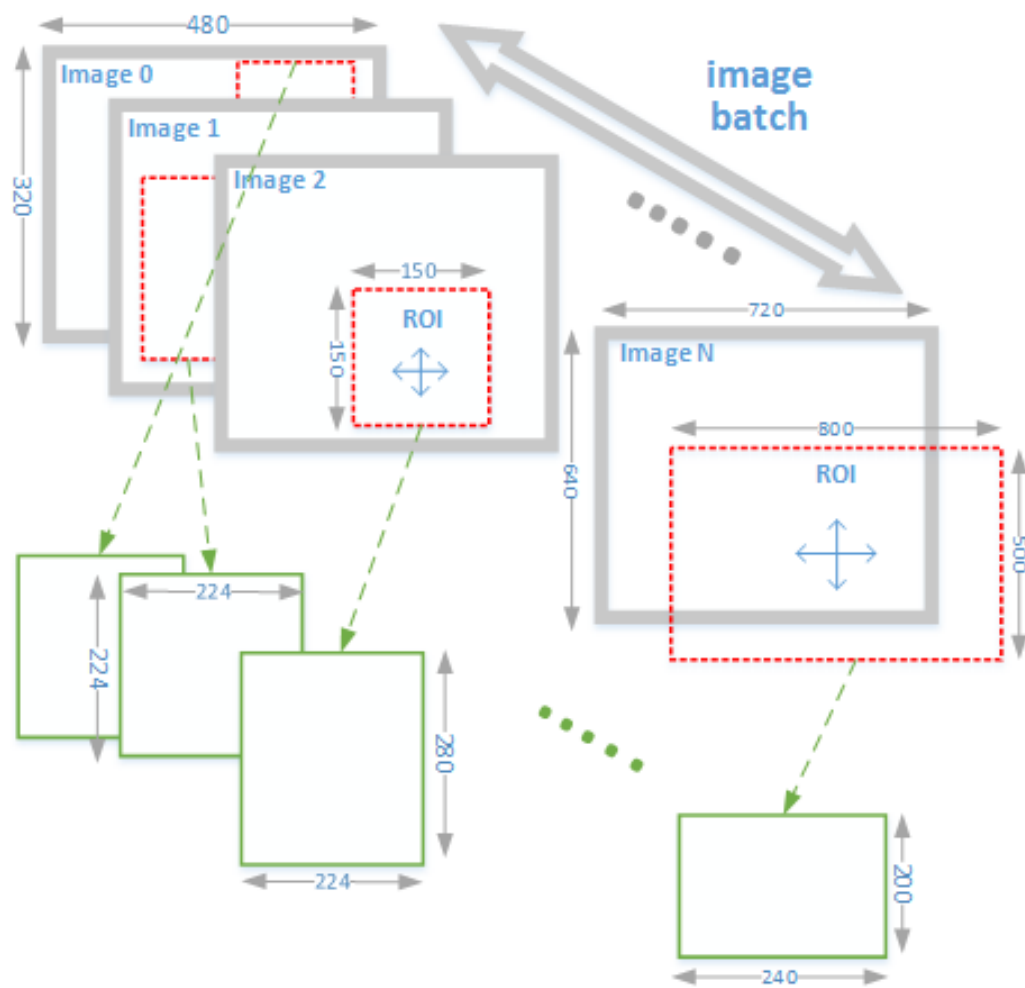
1.11.11. ResizeBatch

Above is the diagram of batch resize functions for variable ROIs. Figure shows the flexibility that the API can handle. The ROIs for source and destination images can be any rectangular width and height that reflects the needed resize factors, inside or beyond the image boundary.

ResizeBatch functions use scale factor automatically determined by the width and height ratios for each pair of input / output images in provided batches.

In this function as in nppiResize the resize scale factor is automatically determined by the width and height ratios of oSrcRectROI and oDstRectROI. If either of those parameters intersect their respective image sizes then pixels outside the image size width and height will not be processed. Details of the resize operation are described above in the Resize section. ResizeBatch generally takes the same

Diagram of batch resize functions for variable ROIs



parameter list as `Resize` except that there is a list of N instances of those parameters ($N > 1$) and that list is passed in device memory. A convenient data structure is provided that allows for easy initialization of the parameter lists. The only restriction on these functions is that there is one single source ROI rectangle and one single destination ROI rectangle which are applied respectively to each image in the batch. The primary purpose of this function is to provide improved performance for batches of smaller images as long as GPU resources are available. Therefore it is recommended that the function not be used for very large images as there may not be resources available for processing several large images simultaneously.

A single set of `oSrcRectROI` and `oDstRectROI` values are applied to each source image and destination image in the batch in the `nppiResizeBatch` version of the function while per image specific `oSrcRectROI` and `oDstRectROI` values can be used in the `nppiResizeBatch_Advanced` version of the function. Source and destination image sizes may vary but `oSmallestSrcSize` and `oSmallestDstSize` must be set to the smallest source and destination image sizes in the batch. The parameters in the *`NppiResizeBatchCXR`* structure represent the corresponding per-image `nppiResize` parameters for each image in the `nppiResizeBatch` functions and the *`NppImageDescriptor`* and *`NppiResizeBatchROI_Advanced`* structures represent the corresponding per-image `nppiResize` parameters for the `nppiResizeBatch_Advanced` functions. The *`NppiResizeBatchCXR`* or *`NppImageDescriptor`* and *`NppiResizeBatchROI_Advanced`* arrays must be in device memory.

`ResizeBatch` supports the following interpolation modes:

```
NPPI_INTER_NN
NPPI_INTER_LINEAR
NPPI_INTER_CUBIC
NPPI_INTER_SUPER
```

The resize primitives return the following error codes: *`Resize Error Codes`*.

1.11.11.1 Common parameters for `nppiResizeBatch` functions:

1.11.11.2 Common parameters for `nppiResizeBatchAdvanced` functions:

param `oSmallestSrcSize`

Size in pixels of the entire smallest source image width and height, may be from different images.

param `oSrcRectROI`

Region of interest in the source images (may overlap source image size width and height).

param `oSmallestDstSize`

Size in pixels of the entire smallest destination image width and height, may be from different images.

param `oDstRectROI`

Region of interest in the destination images (may overlap destination image size width and height).

param eInterpolation

The type of eInterpolation to perform resampling. Currently limited to NPPI_INTER_NN, NPPI_INTER_LINEAR, NPPI_INTER_CUBIC, or NPPI_INTER_SUPER.

param pBatchList

Device memory pointer to nBatchSize list of *NppiResizeBatchCXR* structures.

param pBatchSrc

Device pointer to *NppImageDescriptor* list of source image descriptors. User needs to initialize this structure and copy it to device.

param pBatchDst

Device pointer to *NppImageDescriptor* list of destination image descriptors. User needs to initialize this structure and copy it to device.

param pBatchROI

Device pointer to *NppiResizeBatchROI_Advanced* list of per-image variable ROIs. User needs to initialize this structure and copy it to device.

param nBatchSize

Number of *NppiResizeBatchCXR* structures in this call (must be > 1).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Resize Error Codes

param nMaxWidth

The maximum width of all destination ROIs

param nMaxHeight

The maximum height of all destination ROIs

param pBatchSrc

Device pointer to *NppImageDescriptor* list of source image descriptors. User needs to initialize this structure and copy it to device.

param pBatchDst

Device pointer to *NppImageDescriptor* list of destination image descriptors. User needs to initialize this structure and copy it to device.

param pBatchROI

Device pointer to *NppiResizeBatchROI_Advanced* list of per-image variable ROIs. User needs to initialize this structure and copy it to device.

param nBatchSize

Number of images in a batch.

param eInterpolation

The type of eInterpolation to perform resampling.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Resize Error Codes

Functions

NppStatus nppiResizeBatch_8u_C1R_Ctx(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiSize* oSmallestDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppiResizeBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 8-bit image resize batch.

For common parameter descriptions, see *Common parameters for nppiResizeBatch functions*:

NppStatus nppiResizeBatch_8u_C3R_Ctx(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiSize* oSmallestDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppiResizeBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 8-bit image resize batch.

For common parameter descriptions, see *Common parameters for nppiResizeBatch functions*:

NppStatus nppiResizeBatch_8u_C4R_Ctx(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiSize* oSmallestDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppiResizeBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 8-bit image resize batch.

For common parameter descriptions, see *Common parameters for nppiResizeBatch functions*:

NppStatus nppiResizeBatch_8u_AC4R_Ctx(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiSize* oSmallestDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppiResizeBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 8-bit image resize batch not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResizeBatch functions*:

NppStatus nppiResizeBatch_32f_C1R_Ctx(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiSize* oSmallestDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppiResizeBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image resize batch.

For common parameter descriptions, see *Common parameters for nppiResizeBatch functions*:

NppStatus nppiResizeBatch_32f_C3R_Ctx(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiSize* oSmallestDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppiResizeBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image resize batch.

For common parameter descriptions, see *Common parameters for nppiResizeBatch functions*:

NppStatus **nppiResizeBatch_32f_C4R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiSize* oSmallestDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppiResizeBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image resize batch.

For common parameter descriptions, see *Common parameters for nppiResizeBatch functions*:

NppStatus **nppiResizeBatch_32f_AC4R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiSize* oSmallestDstSize, *NppiRect* oDstRectROI, int eInterpolation, *NppiResizeBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image resize batch not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResizeBatch functions*:

NppStatus **nppiResizeBatch_8u_C1R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight, *NppiImageDescriptor* *pBatchSrc, *NppiImageDescriptor* *pBatchDst, *NppiResizeBatchROI_Advanced* *pBatchROI, unsigned int nBatchSize, int eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit image resize batch for variable ROI.

For common parameter descriptions, see *Common parameters for nppiResizeBatchAdvanced functions*:

NppStatus **nppiResizeBatch_8u_C3R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight, *NppiImageDescriptor* *pBatchSrc, *NppiImageDescriptor* *pBatchDst, *NppiResizeBatchROI_Advanced* *pBatchROI, unsigned int nBatchSize, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit image resize batch for variable ROI.

For common parameter descriptions, see *Common parameters for nppiResizeBatchAdvanced functions*:

NppStatus **nppiResizeBatch_8u_C4R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight, *NppiImageDescriptor* *pBatchSrc, *NppiImageDescriptor* *pBatchDst, *NppiResizeBatchROI_Advanced* *pBatchROI, unsigned int nBatchSize, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit image resize batch for variable ROI.

For common parameter descriptions, see *Common parameters for nppiResizeBatchAdvanced functions*:

NppStatus **nppiResizeBatch_8u_AC4R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight, *NppiImageDescriptor* *pBatchSrc, *NppiImageDescriptor* *pBatchDst, *NppiResizeBatchROI_Advanced* *pBatchROI, unsigned int nBatchSize, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit image resize batch for variable ROI not affecting alpha.

For common parameter descriptions, see [Common parameters for nppiResizeBatchAdvanced functions](#).

NppStatus **nppiResizeBatch_16f_C1R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight,
NppImageDescriptor *pBatchSrc,
NppImageDescriptor *pBatchDst,
NppiResizeBatchROI_Advanced
 *pBatchROI, unsigned int nBatchSize, int
 eInterpolation, *NppStreamContext*
 nppStreamCtx)

1 channel 16-bit floating point image resize batch for variable ROI.

For common parameter descriptions, see [Common parameters for nppiResizeBatchAdvanced functions](#).

NppStatus **nppiResizeBatch_16f_C3R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight,
NppImageDescriptor *pBatchSrc,
NppImageDescriptor *pBatchDst,
NppiResizeBatchROI_Advanced
 *pBatchROI, unsigned int nBatchSize, int
 eInterpolation, *NppStreamContext*
 nppStreamCtx)

3 channel 16-bit floating point image resize batch for variable ROI.

For common parameter descriptions, see [Common parameters for nppiResizeBatchAdvanced functions](#).

NppStatus **nppiResizeBatch_16f_C4R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight,
NppImageDescriptor *pBatchSrc,
NppImageDescriptor *pBatchDst,
NppiResizeBatchROI_Advanced
 *pBatchROI, unsigned int nBatchSize, int
 eInterpolation, *NppStreamContext*
 nppStreamCtx)

4 channel 16-bit floating point image resize batch for variable ROI.

For common parameter descriptions, see [Common parameters for nppiResizeBatchAdvanced functions](#).

NppStatus **nppiResizeBatch_32f_C1R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight,
NppImageDescriptor *pBatchSrc,
NppImageDescriptor *pBatchDst,
NppiResizeBatchROI_Advanced
 *pBatchROI, unsigned int nBatchSize, int
 eInterpolation, *NppStreamContext*
 nppStreamCtx)

1 channel 32-bit floating point image resize batch for variable ROI.

For common parameter descriptions, see [Common parameters for nppiResizeBatchAdvanced functions](#).

NppStatus **nppiResizeBatch_32f_C3R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight, *NppImageDescriptor* *pBatchSrc, *NppImageDescriptor* *pBatchDst, *NppiResizeBatchROI_Advanced* *pBatchROI, unsigned int nBatchSize, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image resize batch for variable ROI.

For common parameter descriptions, see *Common parameters for nppiResizeBatchAdvanced functions*:

NppStatus **nppiResizeBatch_32f_C4R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight, *NppImageDescriptor* *pBatchSrc, *NppImageDescriptor* *pBatchDst, *NppiResizeBatchROI_Advanced* *pBatchROI, unsigned int nBatchSize, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image resize batch for variable ROI.

For common parameter descriptions, see *Common parameters for nppiResizeBatchAdvanced functions*:

NppStatus **nppiResizeBatch_32f_AC4R_Advanced_Ctx**(int nMaxWidth, int nMaxHeight, *NppImageDescriptor* *pBatchSrc, *NppImageDescriptor* *pBatchDst, *NppiResizeBatchROI_Advanced* *pBatchROI, unsigned int nBatchSize, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image resize batch for variable ROI not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiResizeBatchAdvanced functions*:

1.11.12. Remap

Routines providing remap functionality.

Remap supports the following interpolation modes:

NPPI_INTER_NN NPPI_INTER_LINEAR NPPI_INTER_CUBIC NPPI_INTER_CUBIC2P_BSPLINE
NPPI_INTER_CUBIC2P_CATMULLROM NPPI_INTER_CUBIC2P_BO5C03 NPPI_INTER_LANCZOS

Remap chooses source pixels using pixel coordinates explicitly supplied in two 2D device memory image arrays pointed to by the pXMap and pYMap pointers. The pXMap array contains the X coordinated and the pYMap array contains the Y coordinate of the corresponding source image pixel to use as input. These coordinates are in floating point format so fraction pixel positions can be used. The coordinates of the source pixel to sample are determined as follows:

$nSrcX = pXMap[nDstX, nDstY]$ $nSrcY = pYMap[nDstX, nDstY]$

In the Remap functions below source image clip checking is handled as follows:

If the source pixel fractional x and y coordinates are greater than or equal to oSizeROI.x and less than oSizeROI.x + oSizeROI.width and greater than or equal to oSizeROI.y and less than oSizeROI.y + oSizeROI.height then the source pixel is considered to be within the source image clip rectangle and the source image is sampled. Otherwise the source image is not sampled and a destination pixel is not written to the destination image.

1.11.13. Error Codes

The remap primitives return the following error codes:

- NPP_WRONG_INTERSECTION_ROI_ERROR indicates an error condition if srcROIRect has no intersection with the source image.
- NPP_INTERPOLATION_ERROR if eInterpolation has an illegal value.

1.11.13.1 Common parameters for nppiRemap packed pixel functions:

1.11.13.2 Common parameters for nppiRemap planar pixel functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Size in pixels of the source image.

param oSrcROI

Region of interest in the source image.

param pXMap

Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

param nXMapStep

pXMap image array line step in bytes.

param pYMap

Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

param nYMapStep

pYMap image array line step in bytes.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstSizeROI

Region of interest size in the destination image.

param eInterpolation

The type of eInterpolation to perform resampling.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Error Codes

param pSrc

Source-Planar-Image Pointer Array.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Size in pixels of the source image.

param oSrcROI

Region of interest in the source image.

param pXMap

Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

param nXMapStep

pXMap image array line step in bytes.

param pYMap

Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

param nYMapStep

pYMap image array line step in bytes.

param pDst

Destination-Planar-Image Pointer Array.

param nDstStep

Destination-Image Line Step.

param oDstSizeROI

Region of interest size in the destination image.

param eInterpolation

The type of eInterpolation to perform resampling.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Error Codes

Remap

Remaps images.

NppStatus nppiRemap_8u_C1R_Ctx(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

NppStatus nppiRemap_8u_C3R_Ctx(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

NppStatus nppiRemap_8u_C4R_Ctx(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

NppStatus nppiRemap_8u_AC4R_Ctx(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image remap not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

NppStatus nppiRemap_8u_P3R_Ctx(const *Npp8u* *const pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp8u* *pDst[3], int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned planar image remap.

For common parameter descriptions, see *Common parameters for nppiRemap planar pixel functions*.

NppStatus nppiRemap_8u_P4R_Ctx(const *Npp8u* *const pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp8u* *pDst[4], int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned planar image remap.

For common parameter descriptions, see *Common parameters for nppiRemap planar pixel functions*.

```
NppStatus nppiRemap_16u_C1R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int nSrcStep,  
                                NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep,  
                                const Npp32f *pYMap, int nYMapStep, Npp16u *pDst, int  
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

1 channel 16-bit unsigned image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

```
NppStatus nppiRemap_16u_C3R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int nSrcStep,  
                                NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep,  
                                const Npp32f *pYMap, int nYMapStep, Npp16u *pDst, int  
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

3 channel 16-bit unsigned image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

```
NppStatus nppiRemap_16u_C4R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int nSrcStep,  
                                NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep,  
                                const Npp32f *pYMap, int nYMapStep, Npp16u *pDst, int  
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

4 channel 16-bit unsigned image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

```
NppStatus nppiRemap_16u_AC4R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int nSrcStep,  
                                NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep,  
                                const Npp32f *pYMap, int nYMapStep, Npp16u *pDst, int  
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

4 channel 16-bit unsigned image remap not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

```
NppStatus nppiRemap_16u_P3R_Ctx(const Npp16u *const pSrc[3], NppiSize oSrcSize, int nSrcStep,  
                                NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep,  
                                const Npp32f *pYMap, int nYMapStep, Npp16u *pDst[3], int  
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,  
                                NppStreamContext nppStreamCtx)
```

3 channel 16-bit unsigned planar image remap.

For common parameter descriptions, see *Common parameters for nppiRemap planar pixel functions*.

NppStatus nppiRemap_16u_P4R_Ctx(const *Npp16u* *const pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp16u* *pDst[4], int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned planar image remap.

For common parameter descriptions, see *Common parameters for nppiRemap planar pixel functions*.

NppStatus nppiRemap_16s_C1R_Ctx(const *Npp16s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

NppStatus nppiRemap_16s_C3R_Ctx(const *Npp16s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

NppStatus nppiRemap_16s_C4R_Ctx(const *Npp16s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

NppStatus nppiRemap_16s_AC4R_Ctx(const *Npp16s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed image remap not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

NppStatus nppiRemap_16s_P3R_Ctx(const *Npp16s* *const pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp16s* *pDst[3], int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed planar image remap.

For common parameter descriptions, see *Common parameters for nppiRemap planar pixel functions:*.

NppStatus nppiRemap_16s_P4R_Ctx(const *Npp16s* *const pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp16s* *pDst[4], int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed planar image remap.

For common parameter descriptions, see *Common parameters for nppiRemap planar pixel functions:*.

NppStatus nppiRemap_32f_C1R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions:*.

NppStatus nppiRemap_32f_C3R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions:*.

NppStatus nppiRemap_32f_C4R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions:*.

NppStatus nppiRemap_32f_AC4R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image remap not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions:*.

NppStatus nppiRemap_32f_P3R_Ctx(const *Npp32f* *const pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const *Npp32f* *pXMap, int nXMapStep, const *Npp32f* *pYMap, int nYMapStep, *Npp32f* *pDst[3], int nDstStep, *NppiSize* oDstSizeROI, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point planar image remap.

For common parameter descriptions, see *Common parameters for nppiRemap planar pixel functions*.

```
NppStatus nppiRemap_32f_P4R_Ctx(const Npp32f *const pSrc[4], NppiSize oSrcSize, int nSrcStep,
                                NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep,
                                const Npp32f *pYMap, int nYMapStep, Npp32f *pDst[4], int
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,
                                NppStreamContext nppStreamCtx)
```

4 channel 32-bit floating point planar image remap.

For common parameter descriptions, see *Common parameters for nppiRemap planar pixel functions*.

```
NppStatus nppiRemap_64f_C1R_Ctx(const Npp64f *pSrc, NppiSize oSrcSize, int nSrcStep,
                                NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep,
                                const Npp64f *pYMap, int nYMapStep, Npp64f *pDst, int
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,
                                NppStreamContext nppStreamCtx)
```

1 channel 64-bit floating point image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

```
NppStatus nppiRemap_64f_C3R_Ctx(const Npp64f *pSrc, NppiSize oSrcSize, int nSrcStep,
                                NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep,
                                const Npp64f *pYMap, int nYMapStep, Npp64f *pDst, int
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,
                                NppStreamContext nppStreamCtx)
```

3 channel 64-bit floating point image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

```
NppStatus nppiRemap_64f_C4R_Ctx(const Npp64f *pSrc, NppiSize oSrcSize, int nSrcStep,
                                NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep,
                                const Npp64f *pYMap, int nYMapStep, Npp64f *pDst, int
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,
                                NppStreamContext nppStreamCtx)
```

4 channel 64-bit floating point image remap.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.

```
NppStatus nppiRemap_64f_AC4R_Ctx(const Npp64f *pSrc, NppiSize oSrcSize, int nSrcStep,
                                NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep,
                                const Npp64f *pYMap, int nYMapStep, Npp64f *pDst, int
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,
                                NppStreamContext nppStreamCtx)
```

4 channel 64-bit floating point image remap not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiRemap packed pixel functions*.


```
NppStatus nppiRemap_64f_P3R_Ctx(const Npp64f *const pSrc[3], NppiSize oSrcSize, int nSrcStep,
                                NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep,
                                const Npp64f *pYMap, int nYMapStep, Npp64f *pDst[3], int
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,
                                NppStreamContext nppStreamCtx)
```

3 channel 64-bit floating point planar image remap.

For common parameter descriptions, see *Common parameters for nppiRemap planar pixel functions:*.

```
NppStatus nppiRemap_64f_P4R_Ctx(const Npp64f *const pSrc[4], NppiSize oSrcSize, int nSrcStep,
                                NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep,
                                const Npp64f *pYMap, int nYMapStep, Npp64f *pDst[4], int
                                nDstStep, NppiSize oDstSizeROI, int eInterpolation,
                                NppStreamContext nppStreamCtx)
```

4 channel 64-bit floating point planar image remap.

For common parameter descriptions, see *Common parameters for nppiRemap planar pixel functions:*.

1.11.14. Rotate

Rotates an image around the origin (0,0) and then shifts it.

1.11.15. Rotate Error Codes

- ▶ NPP_INTERPOLATION_ERROR if eInterpolation has an illegal value.
- ▶ NPP_RECTANGLE_ERROR Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1.
- ▶ NPP_WRONG_INTERSECTION_ROI_ERROR indicates an error condition if srcROIrect has no intersection with the source image.
- ▶ NPP_WRONG_INTERSECTION_QUAD_WARNING indicates a warning that no operation is performed if the transformed source ROI does not intersect the destination ROI.

1.11.15.1 Common parameters for nppiRotate functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Size in pixels of the source image

param oSrcROI

Region of interest in the source image.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstROI

Region of interest in the destination image.

param nAngle

The angle of rotation in degrees.

param nShiftX

Shift along horizontal axis

param nShiftY

Shift along vertical axis

param eInterpolation

The type of interpolation to perform resampling

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Rotate Error Codes

Rotate

Rotates images.

***NppStatus* nppiRotate_8u_C1R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

8-bit unsigned image rotate.

For common parameter descriptions, see *Common parameters for nppiRotate functions:*

***NppStatus* nppiRotate_8u_C3R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned image rotate.

For common parameter descriptions, see *Common parameters for nppiRotate functions:*

***NppStatus* nppiRotate_8u_C4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image rotate.

For common parameter descriptions, see *Common parameters for nppiRotate functions:*

***NppStatus* nppiRotate_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image rotate ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiRotate functions*:

NppStatus **nppiRotate_16u_C1R_Ctx**(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

16-bit unsigned image rotate.

For common parameter descriptions, see *Common parameters for nppiRotate functions*:

NppStatus **nppiRotate_16u_C3R_Ctx**(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned image rotate.

For common parameter descriptions, see *Common parameters for nppiRotate functions*:

NppStatus **nppiRotate_16u_C4R_Ctx**(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned image rotate.

For common parameter descriptions, see *Common parameters for nppiRotate functions*:

NppStatus **nppiRotate_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned image rotate ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiRotate functions*:

NppStatus **nppiRotate_32f_C1R_Ctx**(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

32-bit float image rotate.

For common parameter descriptions, see *Common parameters for nppiRotate functions*:

NppStatus **nppiRotate_32f_C3R_Ctx**(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

3 channel 32-bit float image rotate.

For common parameter descriptions, see *Common parameters for nppiRotate functions*:

NppStatus **nppiRotate_32f_C4R_Ctx**(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit float image rotate.

For common parameter descriptions, see *Common parameters for nppiRotate functions*:

NppStatus **nppiRotate_32f_AC4R_Ctx**(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit float image rotate ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiRotate functions*.

1.11.16. Rotate Utility Functions

The set of rotate utility functions.

Functions

NppStatus **nppiGetRotateQuad**(*NppiRect* oSrcROI, double aQuad[4][2], double nAngle, double nShiftX, double nShiftY)

Compute shape of rotated image.

Parameters

- ▶ **oSrcROI** – Region-of-interest of the source image.
- ▶ **aQuad** – Host memory array of 2D points. These points are the locations of the corners of the rotated ROI.
- ▶ **nAngle** – The rotation nAngle.
- ▶ **nShiftX** – Post-rotation shift in x-direction
- ▶ **nShiftY** – Post-rotation shift in y-direction

Returns

ROI Related Error Codes.

NppStatus **nppiGetRotateBound**(*NppiRect* oSrcROI, double aBoundingBox[2][2], double nAngle, double nShiftX, double nShiftY)

Compute bounding-box of rotated image.

Parameters

- ▶ **oSrcROI** – Region-of-interest of the source image.
- ▶ **aBoundingBox** – Two 2D points representing the bounding-box of the rotated image. All four points from nppiGetRotateQuad are contained inside the axis-aligned rectangle spanned by the the two points of this bounding box.
- ▶ **nAngle** – The rotation angle.
- ▶ **nShiftX** – Post-rotation shift in x-direction.
- ▶ **nShiftY** – Post-rotation shift in y-direction.

Returns

ROI Related Error Codes.

1.11.17. Mirror

Mirrors images horizontally, vertically or diagonally.

1.11.18. Mirror Error Codes

- ▶ NPP_MIRROR_FLIP_ERROR if flip has an illegal value.
- ▶ NPP_SIZE_ERROR if in_place ROI width or height are not even numbers.

1.11.18.1 Common parameters for nppiMirror functions:

param pSrcDst

In-Place Image Pointer for inplace functions.

param nSrcDstStep

In-Place-Image Line Step for inplace functions.

param pSrc

Source-Image Pointer for non-inplace functions.

param nSrcStep

Source-Image Line Step for non-inplace functions.

param pDst

Destination-Image Pointer for non-inplace functions.

param nDstStep

Destination-Image Line Step for non-inplace functions.

param oROI

Region-Of-Interest (ROI) (in_place ROI widths and heights must be even numbers).

param flip

Specifies the axis about which the image is to be mirrored.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

Mirror

Mirrors images.

NppStatus nppiMirror_8u_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions*.

NppStatus nppiMirror_8u_C1IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions*.

NppStatus nppiMirror_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions*.

NppStatus nppiMirror_8u_C3IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions*.

NppStatus nppiMirror_8u_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions*.

NppStatus nppiMirror_8u_C4IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions*.

NppStatus nppiMirror_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned image mirror not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiMirror functions*.

NppStatus nppiMirror_8u_AC4IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned in place image mirror not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiMirror functions*.

NppStatus nppiMirror_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_16u_C3IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_16u_C4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned image mirror not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_16u_AC4IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned in place image mirror not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_16s_C1IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_16s_C3IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_16s_C4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed image mirror not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_16s_AC4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed in place image mirror not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_32s_C1R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

1 channel 32-bit image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_32s_C1IR_Ctx**(*Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

1 channel 32-bit signed in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_32s_C3R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

3 channel 32-bit image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_32s_C3IR_Ctx**(*Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

3 channel 32-bit signed in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_32s_C4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 32-bit image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_32s_C4IR_Ctx**(*Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_32s_AC4R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 32-bit image mirror not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_32s_AC4IR_Ctx**(*Npp32s* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed in place image mirror not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

1 channel 32-bit float image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_32f_C1IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

1 channel 32-bit float in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

3 channel 32-bit float image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*.

NppStatus **nppiMirror_32f_C3IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

3 channel 32-bit float in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 32-bit float image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_32f_C4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 32-bit float in place image mirror.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 32-bit float image mirror not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

NppStatus **nppiMirror_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oROI, *NppiAxis* flip, *NppStreamContext* nppStreamCtx)

4 channel 32-bit float in place image mirror not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiMirror functions:*

1.11.19. Affine Transforms

1.11.19.1 Affine Transforms

The set of affine transform functions available in the library.

1.11.19.1.1 Affine Transform Error Codes

- **NPP_RECTANGLE_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- **NPP_WRONG_INTERSECTION_ROI_ERROR** Indicates an error condition if oSrcROI has no intersection with the source image
- **NPP_INTERPOLATION_ERROR** Indicates an error condition if interpolation has an illegal value
- **NPP_COEFFICIENT_ERROR** Indicates an error condition if coefficient values are invalid
- **NPP_WRONG_INTERSECTION_QUAD_WARNING** Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI

1.11.19.2 Affine Transform Utility Functions

The set of affine transform utility functions.

Functions

NppStatus **nppiGetAffineTransform**(*NppiRect* oSrcROI, const double aQuad[4][2], double aCoeffs[2][3])

Computes affine transform coefficients based on source ROI and destination quadrilateral.

The function computes the coefficients of an affine transformation that maps the given source ROI (axis aligned rectangle with integer coordinates) to a quadrilateral in the destination image.

An affine transform in 2D is fully determined by the mapping of just three vertices. This function's API allows for passing a complete quadrilateral effectively making the problem overdetermined. What this means in practice is, that for certain quadrilaterals it is not possible to find an affine transform that would map all four corners of the source ROI to the four vertices of that quadrilateral.

The function circumvents this problem by only looking at the first three vertices of the destination image quadrilateral to determine the affine transformation's coefficients. If the destination quadrilateral is indeed one that cannot be mapped using an affine transformation the functions informs the user of this situation by returning a NPP_AFFINE_QUAD_INCORRECT_WARNING.

Parameters

- ▶ **oSrcROI** – The source ROI. This rectangle needs to be at least one pixel wide and high. If either width or height are less than one an NPP_RECTANGLE_ERROR is returned.
- ▶ **aQuad** – Host memory array containing the destination quadrilateral.
- ▶ **aCoeffs** – Host memory array containing the resulting affine transform coefficients.

Returns

Error codes:

- ▶ NPP_SIZE_ERROR Indicates an error condition if any image dimension has zero or negative value
- ▶ NPP_RECTANGLE_ERROR Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- ▶ NPP_COEFFICIENT_ERROR Indicates an error condition if coefficient values are invalid
- ▶ NPP_AFFINE_QUAD_INCORRECT_WARNING Indicates a warning when quad does not conform to the transform properties. Fourth vertex is ignored, internally computed coordinates are used instead

NppStatus **nppiGetAffineQuad**(*NppiRect* oSrcROI, double aQuad[4][2], const double aCoeffs[2][3])

Compute shape of transformed image.

This method computes the quadrilateral in the destination image that the source ROI is transformed into by the affine transformation expressed by the coefficients array (aCoeffs).

Parameters

- ▶ **oSrcROI** – The source ROI.
- ▶ **aQuad** – Host memory array that receives the resulting destination quadrangle.
- ▶ **aCoeffs** – Host memory array containing the affine transform coefficients.

Returns

Error codes:

- ▶ **NPP_SIZE_ERROR** Indicates an error condition if any image dimension has zero or negative value
- ▶ **NPP_RECTANGLE_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- ▶ **NPP_COEFFICIENT_ERROR** Indicates an error condition if coefficient values are invalid

NppStatus **nppiGetAffineBound**(*NppiRect* oSrcROI, double aBound[2][2], const double aCoeffs[2][3])

Compute bounding-box of transformed image.

The method effectively computes the bounding box (axis aligned rectangle) of the transformed source ROI (see *nppiGetAffineQuad()*).

Parameters

- ▶ **oSrcROI** – The source ROI.
- ▶ **aBound** – The resulting bounding box.
- ▶ **aCoeffs** – Host memory array containing the affine transform coefficients.

Returns

Error codes:

- ▶ **NPP_SIZE_ERROR** Indicates an error condition if any image dimension has zero or negative value
- ▶ **NPP_RECTANGLE_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- ▶ **NPP_COEFFICIENT_ERROR** Indicates an error condition if coefficient values are invalid

1.11.19.3 Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is given as a 2×3 matrix (**C**). A pixel location (x, y) in the source image is mapped to (x', y') in the destination image. The destination coordinates are:

$$\begin{aligned} x' &= c_{00} x + c_{01} y + c_{02} \\ y' &= c_{10} x + c_{11} y + c_{12} \end{aligned}$$

The transform matrix is

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \end{bmatrix}$$

Affine transforms decompose into a linear part and a shift. The 2×2 matrix

$$L = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix}$$

represents the linear transform, and the vector

$$v = \begin{bmatrix} c_{02} \\ c_{12} \end{bmatrix}$$

is the post-transform translation; i.e., after applying L , translate by v .

1.11.19.3.1 Common parameters for nppiWarpAffine packed pixel functions:

1.11.19.3.2 Common parameters for nppiWarpAffine planar pixel functions:

param pSrc

Source-Image Pointer.

param oSrcSize

Size of source image in pixels.

param nSrcStep

Source-Image Line Step.

param oSrcROI

Source ROI.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI.

param aCoeffs

Host memory array containing the affine transform coefficients.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Affine Transform Error Codes

param pSrc

Source-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param oSrcSize

Size of source image in pixels.

param nSrcStep

Source-Image Line Step.

param oSrcROI

Source ROI.

param pDst

Destination-Planar-Image Pointer Array. (host memory array containing device memory image plane pointers)

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI.

param aCoeffs

Host memory array containing the affine transform coefficients.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Affine Transform Error Codes

Functions

NppStatus **nppiWarpAffine_8u_C1R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions:*.

NppStatus **nppiWarpAffine_8u_C3R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions:*.

```
NppStatus nppiWarpAffine_8u_C4R_Ctx(const Npp8u *pSrc, NppiSize oSrcSize, int nSrcStep,
                                     NppiRect oSrcROI, Npp8u *pDst, int nDstStep, NppiRect
                                     oDstROI, const double aCoeffs[2][3], int eInterpolation,
                                     NppStreamContext nppStreamCtx)
```

Four-channel 8-bit unsigned affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_8u_AC4R_Ctx(const Npp8u *pSrc, NppiSize oSrcSize, int nSrcStep,
                                       NppiRect oSrcROI, Npp8u *pDst, int nDstStep,
                                       NppiRect oDstROI, const double aCoeffs[2][3], int
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Four-channel 8-bit unsigned affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_8u_P3R_Ctx(const Npp8u *pSrc[3], NppiSize oSrcSize, int nSrcStep,
                                      NppiRect oSrcROI, Npp8u *pDst[3], int nDstStep,
                                      NppiRect oDstROI, const double aCoeffs[2][3], int
                                      eInterpolation, NppStreamContext nppStreamCtx)
```

Three-channel planar 8-bit unsigned affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine planar pixel functions*.

```
NppStatus nppiWarpAffine_8u_P4R_Ctx(const Npp8u *pSrc[4], NppiSize oSrcSize, int nSrcStep,
                                      NppiRect oSrcROI, Npp8u *pDst[4], int nDstStep,
                                      NppiRect oDstROI, const double aCoeffs[2][3], int
                                      eInterpolation, NppStreamContext nppStreamCtx)
```

Four-channel planar 8-bit unsigned affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine planar pixel functions*.

```
NppStatus nppiWarpAffine_16u_C1R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int nSrcStep,
                                       NppiRect oSrcROI, Npp16u *pDst, int nDstStep,
                                       NppiRect oDstROI, const double aCoeffs[2][3], int
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Single-channel 16-bit unsigned affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_16u_C3R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int nSrcStep,
                                       NppiRect oSrcROI, Npp16u *pDst, int nDstStep,
                                       NppiRect oDstROI, const double aCoeffs[2][3], int
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Three-channel 16-bit unsigned affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_16u_C4R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int nSrcStep,
                                       NppiRect oSrcROI, Npp16u *pDst, int nDstStep,
                                       NppiRect oDstROI, const double aCoeffs[2][3], int
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_16u_AC4R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int nSrcStep,
                                       NppiRect oSrcROI, Npp16u *pDst, int nDstStep,
                                       NppiRect oDstROI, const double aCoeffs[2][3], int
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_16u_P3R_Ctx(const Npp16u *pSrc[3], NppiSize oSrcSize, int
                                       nSrcStep, NppiRect oSrcROI, Npp16u *pDst[3], int
                                       nDstStep, NppiRect oDstROI, const double
                                       aCoeffs[2][3], int eInterpolation, NppStreamContext
                                       nppStreamCtx)
```

Three-channel planar 16-bit unsigned affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine planar pixel functions*.

```
NppStatus nppiWarpAffine_16u_P4R_Ctx(const Npp16u *pSrc[4], NppiSize oSrcSize, int
                                       nSrcStep, NppiRect oSrcROI, Npp16u *pDst[4], int
                                       nDstStep, NppiRect oDstROI, const double
                                       aCoeffs[2][3], int eInterpolation, NppStreamContext
                                       nppStreamCtx)
```

Four-channel planar 16-bit unsigned affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine planar pixel functions*.

```
NppStatus nppiWarpAffine_32s_C1R_Ctx(const Npp32s *pSrc, NppiSize oSrcSize, int nSrcStep,
                                       NppiRect oSrcROI, Npp32s *pDst, int nDstStep,
                                       NppiRect oDstROI, const double aCoeffs[2][3], int
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Single-channel 32-bit signed affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_32s_C3R_Ctx(const Npp32s *pSrc, NppiSize oSrcSize, int nSrcStep,
                                       NppiRect oSrcROI, Npp32s *pDst, int nDstStep,
                                       NppiRect oDstROI, const double aCoeffs[2][3], int
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Three-channel 32-bit signed affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_32s_C4R_Ctx(const Npp32s *pSrc, NppiSize oSrcSize, int nSrcStep,
                                       NppiRect oSrcROI, Npp32s *pDst, int nDstStep,
                                       NppiRect oDstROI, const double aCoeffs[2][3], int
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Four-channel 32-bit signed affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*:

NppStatus nppiWarpAffine_32s_AC4R_Ctx(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*:

NppStatus nppiWarpAffine_32s_P3R_Ctx(const *Npp32s* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32s* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 32-bit signed affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine planar pixel functions*:

NppStatus nppiWarpAffine_32s_P4R_Ctx(const *Npp32s* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32s* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 32-bit signed affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine planar pixel functions*:

NppStatus nppiWarpAffine_16f_C1R_Ctx(const *Npp16f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*:

NppStatus nppiWarpAffine_16f_C3R_Ctx(const *Npp16f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*:

NppStatus nppiWarpAffine_16f_C4R_Ctx(const *Npp16f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*:

NppStatus nppiWarpAffine_32f_C1R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

NppStatus nppiWarpAffine_32f_C3R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

NppStatus nppiWarpAffine_32f_C4R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

NppStatus nppiWarpAffine_32f_AC4R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

NppStatus nppiWarpAffine_32f_P3R_Ctx(const *Npp32f* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 32-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine planar pixel functions*.

NppStatus nppiWarpAffine_32f_P4R_Ctx(const *Npp32f* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 32-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine planar pixel functions*.

NppStatus nppiWarpAffine_64f_C1R_Ctx(const *Npp64f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp64f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 64-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_64f_C3R_Ctx(const Npp64f *pSrc, NppiSize oSrcSize, int nSrcStep,  
                                     NppiRect oSrcROI, Npp64f *pDst, int nDstStep,  
                                     NppiRect oDstROI, const double aCoeffs[2][3], int  
                                     eInterpolation, NppStreamContext nppStreamCtx)
```

Three-channel 64-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_64f_C4R_Ctx(const Npp64f *pSrc, NppiSize oSrcSize, int nSrcStep,  
                                     NppiRect oSrcROI, Npp64f *pDst, int nDstStep,  
                                     NppiRect oDstROI, const double aCoeffs[2][3], int  
                                     eInterpolation, NppStreamContext nppStreamCtx)
```

Four-channel 64-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_64f_AC4R_Ctx(const Npp64f *pSrc, NppiSize oSrcSize, int nSrcStep,  
                                       NppiRect oSrcROI, Npp64f *pDst, int nDstStep,  
                                       NppiRect oDstROI, const double aCoeffs[2][3], int  
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Four-channel 64-bit floating-point affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffine packed pixel functions*.

```
NppStatus nppiWarpAffine_64f_P3R_Ctx(const Npp64f *aSrc[3], NppiSize oSrcSize, int nSrcStep,  
                                       NppiRect oSrcROI, Npp64f *aDst[3], int nDstStep,  
                                       NppiRect oDstROI, const double aCoeffs[2][3], int  
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Three-channel planar 64-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine planar pixel functions*.

```
NppStatus nppiWarpAffine_64f_P4R_Ctx(const Npp64f *aSrc[4], NppiSize oSrcSize, int nSrcStep,  
                                       NppiRect oSrcROI, Npp64f *aDst[4], int nDstStep,  
                                       NppiRect oDstROI, const double aCoeffs[2][3], int  
                                       eInterpolation, NppStreamContext nppStreamCtx)
```

Four-channel planar 64-bit floating-point affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffine planar pixel functions*.

1.11.19.4 Affine Transform Batch

Details of the warp affine operation are described above in the WarpAffine section. WarpAffineBatch generally takes the same parameter list as WarpAffine except that there is a list of N instances of those parameters ($N > 1$) and that list is passed in device memory. A convenient data structure is provided that allows for easy initialization of the parameter lists. The aTransformedCoeffs array is for internal use only and should not be directly initialized by the application. The only restriction on these functions is that there is one single source ROI rectangle and one single destination ROI rectangle which are applied respectively to each image in the batch. The primary purpose of this function is to provide improved performance for batches of smaller images as long as GPU resources are available. Therefore it is recommended that the function not be used for very large images as there may not be resources available for processing several large images simultaneously.

A single set of oSrcRectROI and oDstRectROI values are applied to each source image and destination image in the batch. Source and destination image sizes may vary but oSmallestSrcSize must be set to the smallest source and image size in the batch. The parameters in the *NppiWarpAffineBatchCXR* structure represent the corresponding per-image nppiWarpAffine parameters for each image in the batch. The *NppiWarpAffineBatchCXR* array must be in device memory. The nppiWarpAffineBatchInit function MUST be called AFTER the application has initialized the array of *NppiWarpAffineBatchCXR* structures and BEFORE calling any of the nppiWarpAffineBatch functions so that the aTransformedCoeffs array can be internally pre-initialized for each image in the batch. The batch size passed to nppiWarpAffineBatchInit must match the batch size passed to the corresponding warp affine batch function.

WarpAffineBatch supports the following interpolation modes:

```
NPPI_INTER_NN
NPPI_INTER_LINEAR
NPPI_INTER_CUBIC
```

1.11.19.4.1 Codes

The warp affine primitives return the following error codes:

- NPP_RECTANGLE_ERROR if either destination ROI width or height is less than 1 pixel.
- NPP_INTERPOLATION_ERROR if eInterpolation has an illegal value.
- NPP_SIZE_ERROR if source size width or height is less than 2 pixels.

1.11.19.5 Common parameters for nppiWarpAffineBatch functions:

param oSmallestSrcSize

Size in pixels of the entire smallest source image width and height, may be from different images.

param oSrcRectROI

Region of interest in the source images.

param oDstRectROI

Region of interest in the destination images.

param eInterpolation

The type of eInterpolation to perform resampling. Currently limited to NPPI_INTER_NN, NPPI_INTER_LINEAR, or NPPI_INTER_CUBIC.

param pBatchList

Device memory pointer to nBatchSize list of *NppiWarpAffineBatchCXR* structures.

param nBatchSize

Number of *NppiWarpAffineBatchCXR* structures in this call (must be > 1).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiWarpAffineBatchInit_Ctx**(*NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

Initializes the aTransformedCoeffs array in pBatchList for each image in the list.

MUST be called before calling the corresponding warp affine batch function whenever any of the transformation matrices in the list have changed.

Parameters

- ▶ **pBatchList** – Device memory pointer to nBatchSize list of *NppiWarpAffineBatchCXR* structures.
- ▶ **nBatchSize** – Number of *NppiWarpAffineBatchCXR* structures in this call (must be > 1).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

NppStatus **nppiWarpAffineBatch_8u_C1R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer image warp affine batch.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions:*.

NppStatus **nppiWarpAffineBatch_8u_C3R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer image warp affine batch.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions:*.

NppStatus **nppiWarpAffineBatch_8u_C4R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer image warp affine batch.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions*.

NppStatus **nppiWarpAffineBatch_8u_AC4R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer image warp affine batch not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions*.

NppStatus **nppiWarpAffineBatch_16f_C1R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 16-bit floating point image warp affine batch.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions*.

NppStatus **nppiWarpAffineBatch_16f_C3R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 16-bit floating point image warp affine batch.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions*.

NppStatus **nppiWarpAffineBatch_16f_C4R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point image warp affine batch.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions*.

NppStatus **nppiWarpAffineBatch_32f_C1R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image warp affine batch.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions:*.

NppStatus nppiWarpAffineBatch_32f_C3R_Ctx(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image warp affine batch.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions:*.

NppStatus nppiWarpAffineBatch_32f_C4R_Ctx(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image warp affine batch.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions:*.

NppStatus nppiWarpAffineBatch_32f_AC4R_Ctx(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpAffineBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image warp affine batch not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBatch functions:*.

1.11.19.6 Backwards Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is given as a 2×3 matrix (C). A pixel location (x, y) in the source image is mapped to (x', y') in the destination image. The destination coordinates satisfy:

$$\begin{aligned} x &= c_{00} x' + c_{01} y' + c_{02} \\ y &= c_{10} x' + c_{11} y' + c_{12} \end{aligned}$$

with

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \end{bmatrix}.$$

In other words, given C , the source coordinates are transformed to the destination using the inverse matrix C^{-1} :

$$M = C^{-1} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \end{bmatrix}$$

and the forward mapping is

$$\begin{aligned}x' &= m_{00} x + m_{01} y + m_{02} \\y' &= m_{10} x + m_{11} y + m_{12}\end{aligned}$$

1.11.19.6.1 Common parameters for nppiWarpAffineBack packed pixel functions:

1.11.19.6.2 Common parameters for nppiWarpAffineBack planar pixel functions:

param pSrc

Source-Image Pointer.

param oSrcSize

Size of source image in pixels.

param nSrcStep

Source-Image Line Step.

param oSrcROI

Source ROI.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI.

param aCoeffs

Host memory array containing the affine transform coefficients.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Affine Transform Error Codes

param pSrc

Source-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param oSrcSize

Size of source image in pixels.

param nSrcStep

Source-Image Line Step.

param oSrcROI

Source ROI.

param pDst

Destination-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI.

param aCoeffs

Host memory array containing the affine transform coefficients.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Affine Transform Error Codes

Functions

NppStatus **nppiWarpAffineBack_8u_C1R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*

NppStatus **nppiWarpAffineBack_8u_C3R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*

NppStatus **nppiWarpAffineBack_8u_C4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*

NppStatus **nppiWarpAffineBack_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer backwards affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*.

```
NppStatus nppiWarpAffineBack_8u_P3R_Ctx(const Npp8u *pSrc[3], NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp8u *pDst[3], int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Three-channel planar 8-bit unsigned integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack planar pixel functions:*.

```
NppStatus nppiWarpAffineBack_8u_P4R_Ctx(const Npp8u *pSrc[4], NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp8u *pDst[4], int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Four-channel planar 8-bit unsigned integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack planar pixel functions:*.

```
NppStatus nppiWarpAffineBack_16u_C1R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Single-channel 16-bit unsigned integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*.

```
NppStatus nppiWarpAffineBack_16u_C3R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Three-channel 16-bit unsigned integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*.

```
NppStatus nppiWarpAffineBack_16u_C4R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*.

NppStatus **nppiWarpAffineBack_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer backwards affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions*.

NppStatus **nppiWarpAffineBack_16u_P3R_Ctx**(const *Npp16u* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 16-bit unsigned integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack planar pixel functions*.

NppStatus **nppiWarpAffineBack_16u_P4R_Ctx**(const *Npp16u* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 16-bit unsigned integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack planar pixel functions*.

NppStatus **nppiWarpAffineBack_32s_C1R_Ctx**(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit signed integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions*.

NppStatus **nppiWarpAffineBack_32s_C3R_Ctx**(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit signed integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions*.

NppStatus **nppiWarpAffineBack_32s_C4R_Ctx**(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[2][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*.

```
NppStatus nppiWarpAffineBack_32s_AC4R_Ctx(const Npp32s *pSrc, NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Four-channel 32-bit signed integer backwards affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*.

```
NppStatus nppiWarpAffineBack_32s_P3R_Ctx(const Npp32s *pSrc[3], NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp32s *pDst[3], int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Three-channel planar 32-bit signed integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack planar pixel functions:*.

```
NppStatus nppiWarpAffineBack_32s_P4R_Ctx(const Npp32s *pSrc[4], NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp32s *pDst[4], int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Four-channel planar 32-bit signed integer backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack planar pixel functions:*.

```
NppStatus nppiWarpAffineBack_32f_C1R_Ctx(const Npp32f *pSrc, NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp32f *pDst, int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Single-channel 32-bit floating-point backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*.

```
NppStatus nppiWarpAffineBack_32f_C3R_Ctx(const Npp32f *pSrc, NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp32f *pDst, int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Three-channel 32-bit floating-point backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*.

```
NppStatus nppiWarpAffineBack_32f_C4R_Ctx(const Npp32f *pSrc, NppiSize oSrcSize, int
                                         nSrcStep, NppiRect oSrcROI, Npp32f *pDst, int
                                         nDstStep, NppiRect oDstROI, const double
                                         aCoeffs[2][3], int eInterpolation,
                                         NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating-point backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*.

```
NppStatus nppiWarpAffineBack_32f_AC4R_Ctx(const Npp32f *pSrc, NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32f *pDst, int
                                             nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[2][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating-point backwards affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack packed pixel functions:*.

```
NppStatus nppiWarpAffineBack_32f_P3R_Ctx(const Npp32f *pSrc[3], NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32f *pDst[3], int
                                             nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[2][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Three-channel planar 32-bit floating-point backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack planar pixel functions:*.

```
NppStatus nppiWarpAffineBack_32f_P4R_Ctx(const Npp32f *pSrc[4], NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32f *pDst[4], int
                                             nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[2][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Four-channel planar 32-bit floating-point backwards affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineBack planar pixel functions:*.

1.11.19.7 Quad-Based Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is computed such that it maps a quadrilateral in source image space to a quadrilateral in destination image space.

An affine transform is fully determined by the mapping of 3 discrete points. The following primitives compute an affine transformation matrix that maps the first three corners of the source quad are mapped to the first three vertices of the destination image quad. If the fourth vertices do not match the transform, an NPP_AFFINE_QUAD_INCORRECT_WARNING is returned by the primitive.

1.11.19.7.1 Common parameters for nppiWarpAffineQuad packed pixel functions:

1.11.19.7.2 Common parameters for nppiWarpAffineQuad planar pixel functions:

param pSrc

Source-Image Pointer.

param oSrcSize

Size of source image in pixels.

param nSrcStep

Source-Image Line Step.

param oSrcROI

Source ROI.

param aSrcQuad

Source quad.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI.

param aDstQuad

Destination quad.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Affine Transform Error Codes

param pSrc

Source-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param oSrcSize

Size of source image in pixels.

param nSrcStep

Source-Image Line Step.

param oSrcROI

Source ROI

param aSrcQuad

Source quad.

param pDst

Destination-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI

param aDstQuad

Destination quad.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Affine Transform Error Codes

Functions

NppStatus **nppiWarpAffineQuad_8u_C1R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit floating-point quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions:*.

NppStatus **nppiWarpAffineQuad_8u_C3R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions:*.

NppStatus **nppiWarpAffineQuad_8u_C4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions:*.

NppStatus nppiWarpAffineQuad_8u_AC4R_Ctx(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer quad-based affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus nppiWarpAffineQuad_8u_P3R_Ctx(const *Npp8u* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 8-bit unsigned integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad planar pixel functions*.

NppStatus nppiWarpAffineQuad_8u_P4R_Ctx(const *Npp8u* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 8-bit unsigned integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad planar pixel functions*.

NppStatus nppiWarpAffineQuad_16u_C1R_Ctx(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit unsigned integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus nppiWarpAffineQuad_16u_C3R_Ctx(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus **nppiWarpAffineQuad_16u_C4R_Ctx**(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus **nppiWarpAffineQuad_16u_AC4R_Ctx**(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer quad-based affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus **nppiWarpAffineQuad_16u_P3R_Ctx**(const *Npp16u* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 16-bit unsigned integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad planar pixel functions*.

NppStatus **nppiWarpAffineQuad_16u_P4R_Ctx**(const *Npp16u* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 16-bit unsigned integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad planar pixel functions*.

NppStatus **nppiWarpAffineQuad_32s_C1R_Ctx**(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit signed integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus nppiWarpAffineQuad_32s_C3R_Ctx(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit signed integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus nppiWarpAffineQuad_32s_C4R_Ctx(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus nppiWarpAffineQuad_32s_AC4R_Ctx(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed integer quad-based affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus nppiWarpAffineQuad_32s_P3R_Ctx(const *Npp32s* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 32-bit signed integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad planar pixel functions*.

NppStatus nppiWarpAffineQuad_32s_P4R_Ctx(const *Npp32s* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 32-bit signed integer quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad planar pixel functions*.

NppStatus nppiWarpAffineQuad_32f_C1R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit floating-point quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus nppiWarpAffineQuad_32f_C3R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating-point quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus nppiWarpAffineQuad_32f_C4R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus nppiWarpAffineQuad_32f_AC4R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point quad-based affine warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad packed pixel functions*.

NppStatus nppiWarpAffineQuad_32f_P3R_Ctx(const *Npp32f* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 32-bit floating-point quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad planar pixel functions*.

NppStatus **nppiWarpAffineQuad_32f_P4R_Ctx**(const *Npp32f* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 32-bit floating-point quad-based affine warp.

For common parameter descriptions, see *Common parameters for nppiWarpAffineQuad planar pixel functions*.

1.11.20. Perspective Transforms

1.11.20.1 Perspective Transform

The set of perspective transform functions available in the library.

1.11.20.1.1 Perspective Transform Error Codes

- ▶ **NPP_RECTANGLE_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- ▶ **NPP_WRONG_INTERSECTION_ROI_ERROR** Indicates an error condition if oSrcROI has no intersection with the source image
- ▶ **NPP_INTERPOLATION_ERROR** Indicates an error condition if interpolation has an illegal value
- ▶ **NPP_COEFFICIENT_ERROR** Indicates an error condition if coefficient values are invalid
- ▶ **NPP_WRONG_INTERSECTION_QUAD_WARNING** Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI

The set of perspective transform utility functions.

Functions

NppStatus **nppiGetPerspectiveTransform**(*NppiRect* oSrcROI, const double quad[4][2], double aCoeffs[3][3])

Calculates perspective transform coefficients given source rectangular ROI and its destination quadrangle projection.

Parameters

- ▶ **oSrcROI** – Source ROI
- ▶ **quad** – Destination quadrangle
- ▶ **aCoeffs** – Host memory array containing the perspective transform coefficients.

Returns

Error codes:

- ▶ **NPP_SIZE_ERROR** Indicates an error condition if any image dimension has zero or negative value
- ▶ **NPP_RECTANGLE_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- ▶ **NPP_COEFFICIENT_ERROR** Indicates an error condition if coefficient values are invalid

NppStatus **nppiGetPerspectiveQuad**(*NppiRect* oSrcROI, double quad[4][2], const double aCoeffs[3][3])

Calculates perspective transform projection of given source rectangular ROI.

Parameters

- ▶ **oSrcROI** – Source ROI
- ▶ **quad** – Destination quadrangle
- ▶ **aCoeffs** – Host memory array containing the perspective transform coefficients.

Returns

Error codes:

- ▶ **NPP_SIZE_ERROR** Indicates an error condition if any image dimension has zero or negative value
- ▶ **NPP_RECTANGLE_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- ▶ **NPP_COEFFICIENT_ERROR** Indicates an error condition if coefficient values are invalid

NppStatus **nppiGetPerspectiveBound**(*NppiRect* oSrcROI, double bound[2][2], const double aCoeffs[3][3])

Calculates bounding box of the perspective transform projection of the given source rectangular ROI.

Parameters

- ▶ **oSrcROI** – Source ROI
- ▶ **bound** – Bounding box of the transformed source ROI
- ▶ **aCoeffs** – Host memory array containing the perspective transform coefficients.

Returns

Error codes:

- ▶ **NPP_SIZE_ERROR** Indicates an error condition if any image dimension has zero or negative value
- ▶ **NPP_RECTANGLE_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- ▶ **NPP_COEFFICIENT_ERROR** Indicates an error condition if coefficient values are invalid

1.11.20.2 Perspective Transform

Transforms (warps) an image based on a perspective transform.

The perspective (projective) transform is given as a 3×3 matrix (C). A pixel location (x, y) in the source image is mapped to (x', y') in the destination image as:

$$\begin{aligned} x' &= \frac{c_{00} x + c_{01} y + c_{02}}{c_{20} x + c_{21} y + c_{22}}, \\ y' &= \frac{c_{10} x + c_{11} y + c_{12}}{c_{20} x + c_{21} y + c_{22}}. \end{aligned}$$

where

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix}.$$

1.11.20.2.1 Common parameters for nppiWarpPerspective packed pixel functions:

1.11.20.2.2 Common parameters for nppiWarpPerspective planar pixel functions:

Note: The denominator $c_{20} x + c_{21} y + c_{22} \neq 0$ for valid image points.

param pSrc

Source-Image Pointer.

param oSrcSize

Size of source image in pixels.

param nSrcStep

Source-Image Line Step.

param oSrcROI

Source ROI.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI.

param aCoeffs

Host memory array containing the perspective transform coefficients.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

param pSrc

Source-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param oSrcSize

Size of source image in pixels.

param nSrcStep

Source-Image Line Step.

param oSrcROI

Source ROI.

param pDst

Destination-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI.

param aCoeffs

Host memory array containing the perspective transform coefficients.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

Functions

NppStatus **nppiWarpPerspective_8u_C1R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions*.

NppStatus nppiWarpPerspective_8u_C3R_Ctx(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions*.

NppStatus nppiWarpPerspective_8u_C4R_Ctx(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions*.

NppStatus nppiWarpPerspective_8u_AC4R_Ctx(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions*.

NppStatus nppiWarpPerspective_8u_P3R_Ctx(const *Npp8u* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 8-bit unsigned integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective planar pixel functions*.

NppStatus nppiWarpPerspective_8u_P4R_Ctx(const *Npp8u* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 8-bit unsigned integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective planar pixel functions*.

NppStatus nppiWarpPerspective_16u_C1R_Ctx(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit unsigned integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions:*.

```
NppStatus nppiWarpPerspective_16u_C3R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int  
nDstStep, NppiRect oDstROI, const double  
aCoeffs[3][3], int eInterpolation,  
NppStreamContext nppStreamCtx)
```

Three-channel 16-bit unsigned integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions:*.

```
NppStatus nppiWarpPerspective_16u_C4R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int  
nDstStep, NppiRect oDstROI, const double  
aCoeffs[3][3], int eInterpolation,  
NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions:*.

```
NppStatus nppiWarpPerspective_16u_AC4R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int  
nDstStep, NppiRect oDstROI, const double  
aCoeffs[3][3], int eInterpolation,  
NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned integer perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions:*.

```
NppStatus nppiWarpPerspective_16u_P3R_Ctx(const Npp16u *pSrc[3], NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp16u *pDst[3],  
int nDstStep, NppiRect oDstROI, const double  
aCoeffs[3][3], int eInterpolation,  
NppStreamContext nppStreamCtx)
```

Three-channel planar 16-bit unsigned integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective planar pixel functions:*.

```
NppStatus nppiWarpPerspective_16u_P4R_Ctx(const Npp16u *pSrc[4], NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp16u *pDst[4],  
int nDstStep, NppiRect oDstROI, const double  
aCoeffs[3][3], int eInterpolation,  
NppStreamContext nppStreamCtx)
```

Four-channel planar 16-bit unsigned integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective planar pixel functions:*.

```
NppStatus nppiWarpPerspective_32s_C1R_Ctx(const Npp32s *pSrc, NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int  
nDstStep, NppiRect oDstROI, const double  
aCoeffs[3][3], int eInterpolation,  
NppStreamContext nppStreamCtx)
```


Single-channel 32-bit signed integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions:*.

```
NppStatus nppiWarpPerspective_32s_C3R_Ctx(const Npp32s *pSrc, NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int
                                             nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[3][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Three-channel 32-bit signed integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions:*.

```
NppStatus nppiWarpPerspective_32s_C4R_Ctx(const Npp32s *pSrc, NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int
                                             nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[3][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Four-channel 32-bit signed integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions:*.

```
NppStatus nppiWarpPerspective_32s_AC4R_Ctx(const Npp32s *pSrc, NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int
                                             nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[3][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Four-channel 32-bit signed integer perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions:*.

```
NppStatus nppiWarpPerspective_32s_P3R_Ctx(const Npp32s *pSrc[3], NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32s *pDst[3],
                                             int nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[3][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Three-channel planar 32-bit signed integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective planar pixel functions:*.

```
NppStatus nppiWarpPerspective_32s_P4R_Ctx(const Npp32s *pSrc[4], NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32s *pDst[4],
                                             int nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[3][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Four-channel planar 32-bit signed integer perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective planar pixel functions:*.

NppStatus **nppiWarpPerspective_16f_C1R_Ctx**(const *Npp16f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit floating-point perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions*.

NppStatus **nppiWarpPerspective_16f_C3R_Ctx**(const *Npp16f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit floating-point perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions*.

NppStatus **nppiWarpPerspective_16f_C4R_Ctx**(const *Npp16f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit floating-point perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions*.

NppStatus **nppiWarpPerspective_32f_C1R_Ctx**(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit floating-point perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions*.

NppStatus **nppiWarpPerspective_32f_C3R_Ctx**(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating-point perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions*.

NppStatus **nppiWarpPerspective_32f_C4R_Ctx**(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions:*.

```
NppStatus nppiWarpPerspective_32f_AC4R_Ctx(const Npp32f *pSrc, NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32f *pDst, int
                                             nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[3][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating-point perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective packed pixel functions:*.

```
NppStatus nppiWarpPerspective_32f_P3R_Ctx(const Npp32f *pSrc[3], NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32f *pDst[3],
                                             int nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[3][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Three-channel planar 32-bit floating-point perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective planar pixel functions:*.

```
NppStatus nppiWarpPerspective_32f_P4R_Ctx(const Npp32f *pSrc[4], NppiSize oSrcSize, int
                                             nSrcStep, NppiRect oSrcROI, Npp32f *pDst[4],
                                             int nDstStep, NppiRect oDstROI, const double
                                             aCoeffs[3][3], int eInterpolation,
                                             NppStreamContext nppStreamCtx)
```

Four-channel planar 32-bit floating-point perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspective planar pixel functions:*.

1.11.20.3 Perspective Transform Batch

Details of the warp perspective operation are described above in the WarpPerspective section. WarpPerspectiveBatch generally takes the same parameter list as WarpPerspective except that there is a list of N instances of those parameters (N > 1) and that list is passed in device memory. A convenient data structure is provided that allows for easy initialization of the parameter lists. The aTransformedCoeffs array is for internal use only and should not be directly initialized by the application. The only restriction on these functions is that there is one single source ROI rectangle and one single destination ROI rectangle which are applied respectively to each image in the batch. The primary purpose of this function is to provide improved performance for batches of smaller images as long as GPU resources are available. Therefore it is recommended that the function not be used for very large images as there may not be resources available for processing several large images simultaneously.

A single set of oSrcRectROI and oDstRectROI values are applied to each source image and destination image in the batch. Source and destination image sizes may vary but oSmallestSrcSize must be set to the smallest source and image size in the batch. The parameters in the *NppiWarpPerspectiveBatchCXR* structure represent the corresponding per-image nppiWarpPerspective parameters for each image in the batch. The *NppiWarpPerspectiveBatchCXR* array must be in device memory. The nppiWarpPerspectiveBatchInit function MUST be called AFTER the application has initialized the array of *NppiWarpPerspectiveBatchCXR* structures and BEFORE calling any of the nppiWarpPerspectiveBatch functions to so that the aTransformedCoeffs array can be internally pre-initialized for each image in the batch.

The batch size passed to `nppiWarpPerspectiveBatchInit` must match the batch size passed to the corresponding warp perspective batch function.

`WarpPerspectiveBatch` supports the following interpolation modes:

```
NPPI_INTER_NN
NPPI_INTER_LINEAR
NPPI_INTER_CUBIC
```

1.11.20.3.1 Common parameters for `nppiWarpPerspectiveBatch` functions:

param oSmallestSrcSize

Size in pixels of the entire smallest source image width and height, may be from different images.

param oSrcRectROI

Region of interest in the source images.

param oDstRectROI

Region of interest in the destination images.

param eInterpolation

The type of `eInterpolation` to perform resampling. Currently limited to `NPPI_INTER_NN`, `NPPI_INTER_LINEAR`, or `NPPI_INTER_CUBIC`.

param pBatchList

Device memory pointer to `nBatchSize` list of *`NppiWarpPerspectiveBatchCXR`* structures.

param nBatchSize

Number of *`NppiWarpPerspectiveBatchCXR`* structures in this call (must be > 1).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

`NppStatus` **`nppiWarpPerspectiveBatchInit_Ctx`**(*`NppiWarpPerspectiveBatchCXR`* *pBatchList, unsigned int nBatchSize, *`NppStreamContext`* nppStreamCtx)

Initializes the `aTransformedCoeffs` array in `pBatchList` for each image in the list.

MUST be called before calling the corresponding warp perspective batch function whenever any of the transformation matrices in the list have changed.

Parameters

- **pBatchList** – Device memory pointer to `nBatchSize` list of *`NppiWarpPerspectiveBatchCXR`* structures.

- **nBatchSize** – Number of *NppiWarpPerspectiveBatchCXR* structures in this call (must be > 1).
- **nppStreamCtx** – *Application Managed Stream Context*.

NppStatus **nppiWarpPerspectiveBatch_8u_C1R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer image warp perspective batch.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*.

NppStatus **nppiWarpPerspectiveBatch_8u_C3R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer image warp perspective batch.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*.

NppStatus **nppiWarpPerspectiveBatch_8u_C4R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer image warp perspective batch.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*.

NppStatus **nppiWarpPerspectiveBatch_8u_AC4R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer image warp perspective batch not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*.

NppStatus **nppiWarpPerspectiveBatch_16f_C1R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 16-bit floating point image warp perspective batch.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*:

NppStatus **nppiWarpPerspectiveBatch_16f_C3R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 16-bit floating point image warp perspective batch.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*:

NppStatus **nppiWarpPerspectiveBatch_16f_C4R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 16-bit floating point image warp perspective batch.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*:

NppStatus **nppiWarpPerspectiveBatch_32f_C1R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image warp perspective batch.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*:

NppStatus **nppiWarpPerspectiveBatch_32f_C3R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image warp perspective batch.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*:

NppStatus **nppiWarpPerspectiveBatch_32f_C4R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image warp perspective batch.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*:

NppStatus **nppiWarpPerspectiveBatch_32f_AC4R_Ctx**(*NppiSize* oSmallestSrcSize, *NppiRect* oSrcRectROI, *NppiRect* oDstRectROI, int eInterpolation, *NppiWarpPerspectiveBatchCXR* *pBatchList, unsigned int nBatchSize, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image warp perspective batch not affecting alpha.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBatch functions*.

1.11.20.4 Backwards Perspective Transform

Transforms (warps) an image based on a perspective transform.

The perspective (projective) transform is given by a 3×3 matrix (C). A pixel location (x, y) in the source image maps to (x', y') in the destination as:

$$\begin{aligned} x' &= \frac{c_{00}x + c_{01}y + c_{02}}{c_{20}x + c_{21}y + c_{22}}, \\ y' &= \frac{c_{10}x + c_{11}y + c_{12}}{c_{20}x + c_{21}y + c_{22}}. \end{aligned}$$

where

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix}.$$

Conversely, if $M = C^{-1}$, the destination coordinates (x', y') map back to source:

$$\begin{aligned} x &= \frac{m_{00}x' + m_{01}y' + m_{02}}{m_{20}x' + m_{21}y' + m_{22}}, \\ y &= \frac{m_{10}x' + m_{11}y' + m_{12}}{m_{20}x' + m_{21}y' + m_{22}}. \end{aligned}$$

1.11.20.4.1 Common parameters for nppiWarpPerspectiveBack packed pixel functions:

1.11.20.4.2 Common parameters for nppiWarpPerspectiveBack planar pixel functions:

Note: The denominators must be nonzero (valid homogeneous coordinate): $c_{20}x + c_{21}y + c_{22} \neq 0$ and $m_{20}x' + m_{21}y' + m_{22} \neq 0$; also $\det(C) \neq 0$.

param pSrc

Source-Image Pointer.

param oSrcSize

Size of source image in pixels.

param nSrcStep

Source-Image Line Step.

param oSrcROI

Source ROI.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI.

param aCoeffs

Host memory array containing the perspective transform coefficients.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

param pSrc

Source-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param oSrcSize

Size of source image in pixels.

param nSrcStep

Source-Image Line Step.

param oSrcROI

Source ROI.

param pDst

Destination-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI.

param aCoeffs

Host memory array containing the perspective transform coefficients.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes***Functions**

NppStatus **nppiWarpPerspectiveBack_8u_C1R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*.

NppStatus **nppiWarpPerspectiveBack_8u_C3R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*.

NppStatus **nppiWarpPerspectiveBack_8u_C4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*.

NppStatus **nppiWarpPerspectiveBack_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer backwards perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*:

```
NppStatus nppiWarpPerspectiveBack_8u_P3R_Ctx(const Npp8u *pSrc[3], NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp8u  
*pDst[3], int nDstStep, NppiRect oDstROI,  
const double aCoeffs[3][3], int  
eInterpolation, NppStreamContext  
nppStreamCtx)
```

Three-channel planar 8-bit unsigned integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack planar pixel functions*:

```
NppStatus nppiWarpPerspectiveBack_8u_P4R_Ctx(const Npp8u *pSrc[4], NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp8u  
*pDst[4], int nDstStep, NppiRect oDstROI,  
const double aCoeffs[3][3], int  
eInterpolation, NppStreamContext  
nppStreamCtx)
```

Four-channel planar 8-bit unsigned integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack planar pixel functions*:

```
NppStatus nppiWarpPerspectiveBack_16u_C1R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp16u *pDst,  
int nDstStep, NppiRect oDstROI, const  
double aCoeffs[3][3], int eInterpolation,  
NppStreamContext nppStreamCtx)
```

Single-channel 16-bit unsigned integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*:

```
NppStatus nppiWarpPerspectiveBack_16u_C3R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp16u *pDst,  
int nDstStep, NppiRect oDstROI, const  
double aCoeffs[3][3], int eInterpolation,  
NppStreamContext nppStreamCtx)
```

Three-channel 16-bit unsigned integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*:

```
NppStatus nppiWarpPerspectiveBack_16u_C4R_Ctx(const Npp16u *pSrc, NppiSize oSrcSize, int  
nSrcStep, NppiRect oSrcROI, Npp16u *pDst,  
int nDstStep, NppiRect oDstROI, const  
double aCoeffs[3][3], int eInterpolation,  
NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*:

NppStatus nppiWarpPerspectiveBack_16u_AC4R_Ctx(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer backwards perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*.

NppStatus nppiWarpPerspectiveBack_16u_P3R_Ctx(const *Npp16u* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 16-bit unsigned integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack planar pixel functions*.

NppStatus nppiWarpPerspectiveBack_16u_P4R_Ctx(const *Npp16u* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp16u* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 16-bit unsigned integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack planar pixel functions*.

NppStatus nppiWarpPerspectiveBack_32s_C1R_Ctx(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit signed integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*.

NppStatus nppiWarpPerspectiveBack_32s_C3R_Ctx(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit signed integer backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*.

NppStatus nppiWarpPerspectiveBack_32s_C4R_Ctx(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed integer backwards perspective warp.

For common parameter descriptions, see [Common parameters for nppiWarpPerspectiveBack packed pixel functions](#):

```
NppStatus nppiWarpPerspectiveBack_32s_AC4R_Ctx( const Npp32s *pSrc, NppiSize oSrcSize, int
                                                    nSrcStep, NppiRect oSrcROI, Npp32s
                                                    *pDst, int nDstStep, NppiRect oDstROI,
                                                    const double aCoeffs[3][3], int
                                                    eInterpolation, NppStreamContext
                                                    nppStreamCtx )
```

Four-channel 32-bit signed integer backwards perspective warp, ignoring alpha channel.

For common parameter descriptions, see [Common parameters for nppiWarpPerspectiveBack packed pixel functions](#):

```
NppStatus nppiWarpPerspectiveBack_32s_P3R_Ctx( const Npp32s *pSrc[3], NppiSize oSrcSize,
                                                    int nSrcStep, NppiRect oSrcROI, Npp32s
                                                    *pDst[3], int nDstStep, NppiRect oDstROI,
                                                    const double aCoeffs[3][3], int
                                                    eInterpolation, NppStreamContext
                                                    nppStreamCtx )
```

Three-channel planar 32-bit signed integer backwards perspective warp.

For common parameter descriptions, see [Common parameters for nppiWarpPerspectiveBack planar pixel functions](#):

```
NppStatus nppiWarpPerspectiveBack_32s_P4R_Ctx( const Npp32s *pSrc[4], NppiSize oSrcSize,
                                                    int nSrcStep, NppiRect oSrcROI, Npp32s
                                                    *pDst[4], int nDstStep, NppiRect oDstROI,
                                                    const double aCoeffs[3][3], int
                                                    eInterpolation, NppStreamContext
                                                    nppStreamCtx )
```

Four-channel planar 32-bit signed integer backwards perspective warp.

For common parameter descriptions, see [Common parameters for nppiWarpPerspectiveBack planar pixel functions](#):

```
NppStatus nppiWarpPerspectiveBack_32f_C1R_Ctx( const Npp32f *pSrc, NppiSize oSrcSize, int
                                                    nSrcStep, NppiRect oSrcROI, Npp32f *pDst,
                                                    int nDstStep, NppiRect oDstROI, const
                                                    double aCoeffs[3][3], int eInterpolation,
                                                    NppStreamContext nppStreamCtx )
```

Single-channel 32-bit floating-point backwards perspective warp.

For common parameter descriptions, see [Common parameters for nppiWarpPerspectiveBack packed pixel functions](#):

```
NppStatus nppiWarpPerspectiveBack_32f_C3R_Ctx( const Npp32f *pSrc, NppiSize oSrcSize, int
                                                    nSrcStep, NppiRect oSrcROI, Npp32f *pDst,
                                                    int nDstStep, NppiRect oDstROI, const
                                                    double aCoeffs[3][3], int eInterpolation,
                                                    NppStreamContext nppStreamCtx )
```

Three-channel 32-bit floating-point backwards perspective warp.

For common parameter descriptions, see [Common parameters for nppiWarpPerspectiveBack packed pixel functions](#):

NppStatus nppiWarpPerspectiveBack_32f_C4R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*.

NppStatus nppiWarpPerspectiveBack_32f_AC4R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point backwards perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack packed pixel functions*.

NppStatus nppiWarpPerspectiveBack_32f_P3R_Ctx(const *Npp32f* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 32-bit floating-point backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack planar pixel functions*.

NppStatus nppiWarpPerspectiveBack_32f_P4R_Ctx(const *Npp32f* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, *Npp32f* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aCoeffs[3][3], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 32-bit floating-point backwards perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveBack planar pixel functions*.

1.11.20.5 Quad-Based Perspective Transform

Transforms (warps) an image based on an perspective transform.

The perspective transform is computed such that it maps a quadrilateral in source image space to a quadrilateral in destination image space.

1.11.20.5.1 Common parameters for npplWarpPerspectiveQuad packed pixel functions:

1.11.20.5.2 Common parameters for npplWarpPerspectiveQuad planar pixel functions:

param pSrc
Source-Image Pointer.

param oSrcSize
Size of source image in pixels.

param nSrcStep
Source-Image Line Step.

param oSrcROI
Source ROI.

param aSrcQuad
Source quad.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oDstROI
Destination ROI.

param aDstQuad
Destination quad.

param eInterpolation
Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

param pSrc
Source-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param oSrcSize
Size of source image in pixels.

param nSrcStep
Source-Image Line Step.

param oSrcROI
Source ROI.

param aSrcQuad
Source quad.

param pDst

Destination-Planar-Image Pointer Array (host memory array containing device memory image plane pointers).

param nDstStep

Destination-Image Line Step.

param oDstROI

Destination ROI.

param aDstQuad

Destination quad.

param eInterpolation

Interpolation mode: can be NPPI_INTER_NN, NPPI_INTER_LINEAR or NPPI_INTER_CUBIC.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

Functions

NppStatus **nppiWarpPerspectiveQuad_8u_C1R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus **nppiWarpPerspectiveQuad_8u_C3R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus **nppiWarpPerspectiveQuad_8u_C4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus **nppiWarpPerspectiveQuad_8u_AC4R_Ctx**(const *Npp8u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer quad-based perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*:

NppStatus **nppiWarpPerspectiveQuad_8u_P3R_Ctx**(const *Npp8u* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 8-bit unsigned integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad planar pixel functions*:

NppStatus **nppiWarpPerspectiveQuad_8u_P4R_Ctx**(const *Npp8u* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp8u* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 8-bit unsigned integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad planar pixel functions*:

NppStatus **nppiWarpPerspectiveQuad_16u_C1R_Ctx**(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit unsigned integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*:

NppStatus **nppiWarpPerspectiveQuad_16u_C3R_Ctx**(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*:

NppStatus nppiWarpPerspectiveQuad_16u_C4R_Ctx(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus nppiWarpPerspectiveQuad_16u_AC4R_Ctx(const *Npp16u* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer quad-based perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus nppiWarpPerspectiveQuad_16u_P3R_Ctx(const *Npp16u* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 16-bit unsigned integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad planar pixel functions*.

NppStatus nppiWarpPerspectiveQuad_16u_P4R_Ctx(const *Npp16u* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp16u* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 16-bit unsigned integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad planar pixel functions*.

NppStatus nppiWarpPerspectiveQuad_32s_C1R_Ctx(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit signed integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus **nppiWarpPerspectiveQuad_32s_C3R_Ctx**(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit signed integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus **nppiWarpPerspectiveQuad_32s_C4R_Ctx**(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus **nppiWarpPerspectiveQuad_32s_AC4R_Ctx**(const *Npp32s* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed integer quad-based perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus **nppiWarpPerspectiveQuad_32s_P3R_Ctx**(const *Npp32s* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 32-bit signed integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad planar pixel functions*.

NppStatus **nppiWarpPerspectiveQuad_32s_P4R_Ctx**(const *Npp32s* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32s* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 32-bit signed integer quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad planar pixel functions*.

NppStatus nppiWarpPerspectiveQuad_32f_C1R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit floating-point quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus nppiWarpPerspectiveQuad_32f_C3R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating-point quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus nppiWarpPerspectiveQuad_32f_C4R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus nppiWarpPerspectiveQuad_32f_AC4R_Ctx(const *Npp32f* *pSrc, *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst, int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point quad-based perspective warp, ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad packed pixel functions*.

NppStatus nppiWarpPerspectiveQuad_32f_P3R_Ctx(const *Npp32f* *pSrc[3], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst[3], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Three-channel planar 32-bit floating-point quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad planar pixel functions*.

NppStatus **nppiWarpPerspectiveQuad_32f_P4R_Ctx**(const *Npp32f* *pSrc[4], *NppiSize* oSrcSize, int nSrcStep, *NppiRect* oSrcROI, const double aSrcQuad[4][2], *Npp32f* *pDst[4], int nDstStep, *NppiRect* oDstROI, const double aDstQuad[4][2], int eInterpolation, *NppStreamContext* nppStreamCtx)

Four-channel planar 32-bit floating-point quad-based perspective warp.

For common parameter descriptions, see *Common parameters for nppiWarpPerspectiveQuad planar pixel functions*:

1.12. Image Linear Transforms Functions

Linear image transformations.

These functions can be found in the nppist library. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

1.12.1. Fourier Transforms

The set of Fourier transform functions available in the library.

Functions

NppStatus **nppiMagnitude_32fc32f_C1R_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit floating point complex to 32-bit floating point magnitude.

Converts complex-number pixel image to single channel image computing the result pixels as the magnitude of the complex values.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiMagnitudeSqr_32fc32f_C1R_Ctx**(const *Npp32fc* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

32-bit floating point complex to 32-bit floating point squared magnitude.

Converts complex-number pixel image to single channel image computing the result pixels as the squared magnitude of the complex values.

The squared magnitude is an intermediate result of magnitude computation and can thus be computed faster than actual magnitude. If magnitudes are required for sorting/comparing only, using this function instead of `nppiMagnitude_32fc32f_C1R` can be a worthwhile performance optimization.

Parameters

- ▶ **pSrc** – *Source-Image Pointer*.
- ▶ **nSrcStep** – *Source-Image Line Step*.
- ▶ **pDst** – *Destination-Image Pointer*.
- ▶ **nDstStep** – *Destination-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.13. Image Morphological Operations

Morphological image operations.

Morphological operations are classified as *Neighborhood Operations*.

These functions can be found in the `nppim` library. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

1.13.1. Dilation Functions

1.13.1.1 Image Dilate

1.13.1.1.1 Dilation

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask. Pixels who's corresponding mask values are zero do not participate in the maximum search.

It is the user's responsibility to avoid *Sampling Beyond Image Boundaries*.

1.13.1.1.1.1 Common parameters for nppiDilate functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param pMask
Device memory pointer to the start address of the mask array (non-zero values select participating pixels).

param oMaskSize
Width and Height mask array.

param oAnchor
X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiDilate_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer dilation.

For common parameter descriptions, see *Common parameters for nppiDilate functions:*.

NppStatus **nppiDilate_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer dilation.

For common parameter descriptions, see *Common parameters for nppiDilate functions:*.

NppStatus **nppiDilate_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer dilation.

For common parameter descriptions, see *Common parameters for nppiDilate functions:*.

NppStatus nppiDilate_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer dilation, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilate functions*.

NppStatus nppiDilate_16u_C1R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit unsigned integer dilation.

For common parameter descriptions, see *Common parameters for nppiDilate functions*.

NppStatus nppiDilate_16u_C3R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned integer dilation.

For common parameter descriptions, see *Common parameters for nppiDilate functions*.

NppStatus nppiDilate_16u_C4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer dilation.

For common parameter descriptions, see *Common parameters for nppiDilate functions*.

NppStatus nppiDilate_16u_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer dilation, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilate functions*.

NppStatus nppiDilate_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit floating-point dilation.

For common parameter descriptions, see *Common parameters for nppiDilate functions*.

NppStatus nppiDilate_32f_C3R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating-point dilation.

For common parameter descriptions, see *Common parameters for nppiDilate functions*.

NppStatus nppiDilate_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point dilation.

For common parameter descriptions, see *Common parameters for nppiDilate functions*.

NppStatus **nppiDilate_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point dilation, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilate functions*.

1.13.1.2 Image Dilate Border

1.13.1.2.1 Dilation with border control

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum search. For gray scale dilation the mask contains signed mask values which are added to the corresponding source image sample value before determining the maximum value after clamping.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.13.1.2.1.1 Common parameters for nppiDilateBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

Source image starting point relative to pSrc.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pMask

Device memory pointer to the start address of the mask array (non-zero values select participating pixels).

param oMaskSize

Width and Height mask array.

param oAnchor

X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiDilateBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions:*.

NppStatus **nppiDilateBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions:*.

NppStatus **nppiDilateBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions:*.

NppStatus **nppiDilateBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer dilation with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions:*.

NppStatus **nppiDilateBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit unsigned integer dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions*:

```
NppStatus nppiDilateBorder_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u *pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

Three-channel 16-bit unsigned integer dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions*:

```
NppStatus nppiDilateBorder_16u_C4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const Npp8u *pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned integer dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions*:

```
NppStatus nppiDilateBorder_16u_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const Npp8u *pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned integer dilation with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions*:

```
NppStatus nppiDilateBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u *pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

Single-channel 32-bit floating-point dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions*:

```
NppStatus nppiDilateBorder_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u *pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

Three-channel 32-bit floating-point dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions*:

```
NppStatus nppiDilateBorder_32f_C4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp8u *pMask, NppiSize oMaskSize, NppiPoint oAnchor, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating-point dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions*:

NppStatus **nppiDilateBorder_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point dilation with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions*:

NppStatus **nppiGrayDilateBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32s* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer gray scale dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions*:

NppStatus **nppiGrayDilateBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp32f* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit floating point gray scale dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilateBorder functions*:

1.13.1.3 Image Dilate 3x3

1.13.1.3.1 Dilate3x3

Dilation using a 3x3 mask with the anchor at its center pixel.

It is the user's responsibility to avoid *Sampling Beyond Image Boundaries*.

1.13.1.3.1.1 Common parameters for nppiDilate3x3 functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiDilate3x3_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer 3x3 dilation.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus **nppiDilate3x3_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer 3x3 dilation.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus **nppiDilate3x3_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer 3x3 dilation.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus **nppiDilate3x3_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer 3x3 dilation, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus **nppiDilate3x3_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit unsigned integer 3x3 dilation.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus **nppiDilate3x3_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned integer 3x3 dilation.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus nppiDilate3x3_16u_C4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer 3x3 dilation.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus nppiDilate3x3_16u_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer 3x3 dilation, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus nppiDilate3x3_32f_C1R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit floating-point 3x3 dilation.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus nppiDilate3x3_32f_C3R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating-point 3x3 dilation.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus nppiDilate3x3_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point 3x3 dilation.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus nppiDilate3x3_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point 3x3 dilation, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

NppStatus nppiDilate3x3_64f_C1R_Ctx(const *Npp64f* *pSrc, *Npp32s* nSrcStep, *Npp64f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel 64-bit floating-point 3x3 dilation.

For common parameter descriptions, see *Common parameters for nppiDilate3x3 functions*:

1.13.1.4 Image Dilate 3x3 Border

1.13.1.4.1 Dilate3x3Border

Dilation using a 3x3 mask with the anchor at its center pixel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.13.1.4.1.1 Common parameters for nppiDilate3x3Border functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param oSrcSize
Source image width and height in pixels relative to pSrc.

param oSrcOffset
Source image starting point relative to pSrc.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param eBorderType
The border type operation to be applied at source image border boundaries.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiDilate3x3Border_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer 3x3 dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilate3x3Border functions*.

NppStatus nppiDilate3x3Border_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer 3x3 dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilate3x3Border functions*.

NppStatus nppiDilate3x3Border_8u_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer 3x3 dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilate3x3Border functions*.

NppStatus nppiDilate3x3Border_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer 3x3 dilation with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilate3x3Border functions*.

NppStatus nppiDilate3x3Border_16u_C1R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit unsigned integer 3x3 dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilate3x3Border functions*.

NppStatus nppiDilate3x3Border_16u_C3R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned integer 3x3 dilation with border control.

NppStatus nppiDilate3x3Border_16u_C4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer 3x3 dilation with border control.

NppStatus nppiDilate3x3Border_16u_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer 3x3 dilation with border control, ignoring alpha-channel.

```
NppStatus nppiDilate3x3Border_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
                                             oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                             Npp32s nDstStep, NppiSize oSizeROI,
                                             NppiBorderType eBorderType,
                                             NppStreamContext nppStreamCtx)
```

Single-channel 32-bit floating-point 3x3 dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilate3x3Border functions*.

```
NppStatus nppiDilate3x3Border_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
                                             oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                             Npp32s nDstStep, NppiSize oSizeROI,
                                             NppiBorderType eBorderType,
                                             NppStreamContext nppStreamCtx)
```

Three-channel 32-bit floating-point 3x3 dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilate3x3Border functions*.

```
NppStatus nppiDilate3x3Border_32f_C4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize
                                             oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                             int nDstStep, NppiSize oSizeROI, NppiBorderType
                                             eBorderType, NppStreamContext
                                             nppStreamCtx)
```

Four-channel 32-bit floating-point 3x3 dilation with border control.

For common parameter descriptions, see *Common parameters for nppiDilate3x3Border functions*.

```
NppStatus nppiDilate3x3Border_32f_AC4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize
                                             oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                             int nDstStep, NppiSize oSizeROI,
                                             NppiBorderType eBorderType,
                                             NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating-point 3x3 dilation with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiDilate3x3Border functions*.

1.13.2. Erosion Functions

1.13.2.1 Image Erode

1.13.2.1.1 Erode

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask. Pixels who's corresponding mask values are zero do not participate in the maximum search.

It is the user's responsibility to avoid *Sampling Beyond Image Boundaries*.

1.13.2.1.1.1 Common parameters for nppiErode functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param pMask
 Device memory pointer to the start address of the mask array (non-zero values select participating pixels).

param oMaskSize
 Width and Height mask array.

param oAnchor
 X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiErode_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer erosion.

For common parameter descriptions, see *Common parameters for nppiErode functions:*.

NppStatus **nppiErode_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer erosion.

For common parameter descriptions, see *Common parameters for nppiErode functions:*.

NppStatus **nppiErode_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer erosion.

For common parameter descriptions, see *Common parameters for nppiErode functions:*.

NppStatus **nppiErode_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer erosion, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErode functions*.

NppStatus **nppiErode_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit unsigned integer erosion.

For common parameter descriptions, see *Common parameters for nppiErode functions*.

NppStatus **nppiErode_16u_C3R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned integer erosion.

NppStatus **nppiErode_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer erosion.

For common parameter descriptions, see *Common parameters for nppiErode functions*.

NppStatus **nppiErode_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer erosion, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErode functions*.

NppStatus **nppiErode_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit floating-point erosion.

For common parameter descriptions, see *Common parameters for nppiErode functions*.

NppStatus **nppiErode_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating-point erosion.

For common parameter descriptions, see *Common parameters for nppiErode functions*.

NppStatus **nppiErode_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point erosion.

For common parameter descriptions, see *Common parameters for nppiErode functions*.

NppStatus **nppiErode_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point erosion, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErode functions*.

1.13.2.2 Image Erode Border

1.13.2.2.1 Erosion with border control

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the minimum search. For gray scale erosion the mask contains signed mask values which are added to the corresponding source image sample value before determining the minimum value after clamping.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.13.2.2.1.1 Common parameters for nppiErodeBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

Source image starting point relative to pSrc.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pMask

Device memory pointer to the start address of the mask array (non-zero values select participating pixels).

param oMaskSize

Width and Height mask array.

param oAnchor

X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiErodeBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*.

NppStatus **nppiErodeBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*.

NppStatus **nppiErodeBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*.

NppStatus **nppiErodeBorder_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer erosion with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*.

NppStatus **nppiErodeBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit unsigned integer erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*:

```
NppStatus nppiErodeBorder_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst,
Npp32s nDstStep, NppiSize oSizeROI, const Npp8u
*pMask, NppiSize oMaskSize, NppiPoint oAnchor,
NppiBorderType eBorderType, NppStreamContext
nppStreamCtx)
```

Three-channel 16-bit unsigned integer erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*:

```
NppStatus nppiErodeBorder_16u_C4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16u *pDst, int nDstStep,
NppiSize oSizeROI, const Npp8u *pMask, NppiSize
oMaskSize, NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned integer erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*:

```
NppStatus nppiErodeBorder_16u_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp16u *pDst, int nDstStep,
NppiSize oSizeROI, const Npp8u *pMask, NppiSize
oMaskSize, NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned integer erosion with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*:

```
NppStatus nppiErodeBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI, const Npp8u
*pMask, NppiSize oMaskSize, NppiPoint oAnchor,
NppiBorderType eBorderType, NppStreamContext
nppStreamCtx)
```

Single-channel 32-bit floating-point erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*:

```
NppStatus nppiErodeBorder_32f_C3R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
Npp32s nDstStep, NppiSize oSizeROI, const Npp8u
*pMask, NppiSize oMaskSize, NppiPoint oAnchor,
NppiBorderType eBorderType, NppStreamContext
nppStreamCtx)
```

Three-channel 32-bit floating-point erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*:

```
NppStatus nppiErodeBorder_32f_C4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize,
NppiPoint oSrcOffset, Npp32f *pDst, int nDstStep,
NppiSize oSizeROI, const Npp8u *pMask, NppiSize
oMaskSize, NppiPoint oAnchor, NppiBorderType
eBorderType, NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating-point erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*:

```
NppStatus nppiErodeBorder_32f_AC4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize,
                                         NppiPoint oSrcOffset, Npp32f *pDst, int nDstStep,
                                         NppiSize oSizeROI, const Npp8u *pMask, NppiSize
                                         oMaskSize, NppiPoint oAnchor, NppiBorderType
                                         eBorderType, NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating-point erosion with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*.

```
NppStatus nppiGrayErodeBorder_8u_C1R_Ctx(const Npp8u *pSrc, Npp32s nSrcStep, NppiSize
                                           oSrcSize, NppiPoint oSrcOffset, Npp8u *pDst,
                                           Npp32s nDstStep, NppiSize oSizeROI, const
                                           Npp32s *pMask, NppiSize oMaskSize, NppiPoint
                                           oAnchor, NppiBorderType eBorderType,
                                           NppStreamContext nppStreamCtx)
```

Single-channel 8-bit unsigned integer gray scale erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*.

```
NppStatus nppiGrayErodeBorder_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
                                           oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                           Npp32s nDstStep, NppiSize oSizeROI, const
                                           Npp32f *pMask, NppiSize oMaskSize, NppiPoint
                                           oAnchor, NppiBorderType eBorderType,
                                           NppStreamContext nppStreamCtx)
```

Single-channel 32-bit floating point gray scale erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErodeBorder functions*.

1.13.2.3 Image Erode 3x3

1.13.2.3.1 Erode3x3

Erosion using a 3x3 mask with the anchor at its center pixel.

It is the user's responsibility to avoid *Sampling Beyond Image Boundaries*.

1.13.2.3.1.1 Common parameters for nppiErode3x3 functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus nppiErode3x3_8u_C1R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer 3x3 erosion.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*:

NppStatus nppiErode3x3_8u_C3R_Ctx(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer 3x3 erosion.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*:

NppStatus nppiErode3x3_8u_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer 3x3 erosion.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*:

NppStatus nppiErode3x3_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer 3x3 erosion, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*:

NppStatus nppiErode3x3_16u_C1R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel 16-bit unsigned integer 3x3 erosion.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*:

NppStatus nppiErode3x3_16u_C3R_Ctx(const *Npp16u* *pSrc, *Npp32s* nSrcStep, *Npp16u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned integer 3x3 erosion.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*:

NppStatus nppiErode3x3_16u_C4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer 3x3 erosion.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*:

NppStatus **nppiErode3x3_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned integer 3x3 erosion, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*.

NppStatus **nppiErode3x3_32f_C1R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel 32-bit floating-point 3x3 erosion.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*.

NppStatus **nppiErode3x3_32f_C3R_Ctx**(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating-point 3x3 erosion.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*.

NppStatus **nppiErode3x3_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point 3x3 erosion.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*.

NppStatus **nppiErode3x3_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point 3x3 erosion, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*.

NppStatus **nppiErode3x3_64f_C1R_Ctx**(const *Npp64f* *pSrc, *Npp32s* nSrcStep, *Npp64f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Single-channel 64-bit floating-point 3x3 erosion.

For common parameter descriptions, see *Common parameters for nppiErode3x3 functions*.

1.13.2.4 Image Erode 3x3 Border

1.13.2.4.1 Erode3x3Border

Erosion using a 3x3 mask with the anchor at its center pixel with border control.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.13.2.4.1.1 Common parameters for nppiErode3x3Border functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param oSrcSize
 Source image width and height in pixels relative to pSrc.

param oSrcOffset
 Source image starting point relative to pSrc.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param eBorderType
 The border type operation to be applied at source image border boundaries.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiErode3x3Border_8u_C1R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Single-channel 8-bit unsigned integer 3x3 erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions:*.

NppStatus **nppiErode3x3Border_8u_C3R_Ctx**(const *Npp8u* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned integer 3x3 erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions:*.

NppStatus **nppiErode3x3Border_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned integer 3x3 erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions*:

```
NppStatus nppiErode3x3Border_8u_AC4R_Ctx(const Npp8u *pSrc, int nSrcStep, NppiSize
                                         oSrcSize, NppiPoint oSrcOffset, Npp8u *pDst, int
                                         nDstStep, NppiSize oSizeROI, NppiBorderType
                                         eBorderType, NppStreamContext nppStreamCtx)
```

Four-channel 8-bit unsigned integer 3x3 erosion with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions*:

```
NppStatus nppiErode3x3Border_16u_C1R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, NppiSize
                                         oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst,
                                         Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiBorderType eBorderType, NppStreamContext
                                         nppStreamCtx)
```

Single-channel 16-bit unsigned integer 3x3 erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions*:

```
NppStatus nppiErode3x3Border_16u_C3R_Ctx(const Npp16u *pSrc, Npp32s nSrcStep, NppiSize
                                         oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst,
                                         Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiBorderType eBorderType, NppStreamContext
                                         nppStreamCtx)
```

Three-channel 16-bit unsigned integer 3x3 erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions*:

```
NppStatus nppiErode3x3Border_16u_C4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize
                                         oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst, int
                                         nDstStep, NppiSize oSizeROI, NppiBorderType
                                         eBorderType, NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned integer 3x3 erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions*:

```
NppStatus nppiErode3x3Border_16u_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize
                                         oSrcSize, NppiPoint oSrcOffset, Npp16u *pDst,
                                         int nDstStep, NppiSize oSizeROI, NppiBorderType
                                         eBorderType, NppStreamContext
                                         nppStreamCtx)
```

Four-channel 16-bit unsigned integer 3x3 erosion with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions*:

```
NppStatus nppiErode3x3Border_32f_C1R_Ctx(const Npp32f *pSrc, Npp32s nSrcStep, NppiSize
                                         oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                         Npp32s nDstStep, NppiSize oSizeROI,
                                         NppiBorderType eBorderType, NppStreamContext
                                         nppStreamCtx)
```

Single-channel 32-bit floating-point 3x3 erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions*.

NppStatus nppiErode3x3Border_32f_C3R_Ctx(const *Npp32f* *pSrc, *Npp32s* nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, *Npp32s* nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating-point 3x3 erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions*.

NppStatus nppiErode3x3Border_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point 3x3 erosion with border control.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions*.

NppStatus nppiErode3x3Border_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating-point 3x3 erosion with border control, ignoring alpha-channel.

For common parameter descriptions, see *Common parameters for nppiErode3x3Border functions*.

1.13.3. Image Complex Morphological Operations

1.13.3.1 Image Morph

1.13.3.1.1 ComplexImageMorphology

Complex image morphological operations.

1.13.3.2 Image Morph Get Buffer Size

1.13.3.2.1 MorphGetBufferSize

Before calling any of the MorphCloseBorder, MorphOpenBorder, MorphTopHatBorder, MorphBlackHatBorder, or MorphGradientBorder functions the application first needs to call the corresponding MorphGetBufferSize to determine the amount of device memory to allocate as a working buffer. The application allocated device memory is then passed as the pBuffer parameter to the corresponding MorphXXXBorder function.

1.13.3.2.1.1 Common parameters for nppiMorphGetBufferSize functions:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size in bytes.

Functions

NppStatus nppiMorphGetBufferSize_8u_C1R(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for 1 channel 8-bit unsigned integer MorphCloseBorder, MorphOpenBorder, MorphTopHatBorder, MorphBlackHatBorder, or MorphGradientBorder function based on destination image oSizeROI width and height.

For common parameter descriptions, see *Common parameters for nppiMorphGetBufferSize functions*.

NppStatus nppiMorphGetBufferSize_8u_C3R(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for 3 channel 8-bit unsigned integer MorphCloseBorder, MorphOpenBorder, MorphTopHatBorder, MorphBlackHatBorder or MorphGradientBorder function based on destination image oSizeROI width and height.

For common parameter descriptions, see *Common parameters for nppiMorphGetBufferSize functions*.

NppStatus nppiMorphGetBufferSize_8u_C4R(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for 4 channel 8-bit unsigned integer MorphCloseBorder, MorphOpenBorder, MorphTopHatBorder, MorphBlackHatBorder, or MorphGradientBorder function based on destination image oSizeROI width and height.

For common parameter descriptions, see *Common parameters for nppiMorphGetBufferSize functions*.

NppStatus nppiMorphGetBufferSize_16u_C1R(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for 1 channel 16-bit unsigned integer MorphCloseBorder, MorphOpenBorder, MorphTopHatBorder, MorphBlackHatBorder, or MorphGradientBorder function based on destination image oSizeROI width and height.

For common parameter descriptions, see *Common parameters for nppiMorphGetBufferSize functions*.

NppStatus nppiMorphGetBufferSize_16s_C1R(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for 1 channel 16-bit signed integer MorphCloseBorder, MorphOpenBorder, MorphTopHatBorder, MorphBlackHatBorder, or MorphGradientBorder function based on destination image oSizeROI width and height.

For common parameter descriptions, see *Common parameters for nppiMorphGetBufferSize functions*.

NppStatus nppiMorphGetBufferSize_32f_C1R(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for 1 channel 32-bit floating point MorphCloseBorder, MorphOpenBorder, MorphTopHatBorder, MorphBlackHatBorder, or MorphGradientBorder function based on destination image oSizeROI width and height.

For common parameter descriptions, see *Common parameters for nppiMorphGetBufferSize functions*:

NppStatus nppiMorphGetBufferSize_32f_C3R(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for 3 channel 32-bit floating point MorphCloseBorder, MorphOpenBorder, MorphTopHatBorder, MorphBlackHatBorder, or MorphGradientBorder function based on destination image oSizeROI width and height.

For common parameter descriptions, see *Common parameters for nppiMorphGetBufferSize functions*:

NppStatus nppiMorphGetBufferSize_32f_C4R(*NppiSize* oSizeROI, int *hpBufferSize)

Calculate scratch buffer size needed for 4 channel 32-bit floating point MorphCloseBorder, MorphOpenBorder, MorphTopHatBorder, MorphBlackHatBorder, or MorphGradientBorder function based on destination image oSizeROI width and height.

For common parameter descriptions, see *Common parameters for nppiMorphGetBufferSize functions*:

1.13.3.3 Image Morph Close Border

1.13.3.3.1 MorphCloseBorder

Dilation followed by Erosion with border control.

Morphological close computes the output pixel as the maximum pixel value of the pixels under the mask followed by a second pass using the result of the first pass as input which outputs the minimum pixel value of the pixels under the same mask. Pixels whose corresponding mask values are zero do not participate in the maximum or minimum search.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image. The mask is centered over the source image pixel being tested.

Before calling any of the MorphCloseBorder functions the application first needs to call the corresponding MorphGetBufferSize to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding MorphCloseBorder function.

Use the oSrcOffset and oSrcSize parameters to control where the border control operation is applied to the source image ROI borders.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.13.3.3.1.1 Common parameters for nppiMorphCloseBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

Source image starting point relative to pSrc.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pMask

Device memory pointer to the start address of the mask array (non-zero values select participating pixels).

param oMaskSize

Width and Height mask array.

param oAnchor

X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

param pBuffer

Pointer to device memory scratch buffer at least as large as value returned by the corresponding MorphGetBufferSize call.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiMorphCloseBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer morphological close with border control.

For common parameter descriptions, see *Common parameters for nppiMorphCloseBorder functions*.

NppStatus **nppiMorphCloseBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer morphological close with border control.

For common parameter descriptions, see *Common parameters for nppiMorphCloseBorder functions*.

NppStatus **nppiMorphCloseBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer morphological close with border control.

For common parameter descriptions, see *Common parameters for nppiMorphCloseBorder functions*.

NppStatus **nppiMorphCloseBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer morphological close with border control.

For common parameter descriptions, see *Common parameters for nppiMorphCloseBorder functions*.

NppStatus **nppiMorphCloseBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed integer morphological close with border control.

For common parameter descriptions, see *Common parameters for nppiMorphCloseBorder functions*.

NppStatus **nppiMorphCloseBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point morphological close with border control.

For common parameter descriptions, see *Common parameters for nppiMorphCloseBorder functions*.

NppStatus **nppiMorphCloseBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point morphological close with border control.

For common parameter descriptions, see *Common parameters for nppiMorphCloseBorder functions*:

```
NppStatus nppiMorphCloseBorder_32f_C4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize
                                             oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
                                             int nDstStep, NppiSize oSizeROI, const Npp8u
                                             *pMask, NppiSize oMaskSize, NppiPoint
                                             oAnchor, Npp8u *pBuffer, NppiBorderType
                                             eBorderType, NppStreamContext
                                             nppStreamCtx)
```

4 channel 32-bit floating point morphological close with border control.

For common parameter descriptions, see *Common parameters for nppiMorphCloseBorder functions*:

1.13.3.4 Image Morph Open Border

1.13.3.4.1 MorphOpenBorder

Erosion followed by Dilation with border control.

Morphological open computes the output pixel as the minimum pixel value of the pixels under the mask followed by a second pass using the result of the first pass as input which outputs the maximum pixel value of the pixels under the same mask. Pixels whose corresponding mask values are zero do not participate in the minimum or maximum search.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image. The mask is centered over the source image pixel being tested.

Before calling any of the MorphOpenBorder functions the application first needs to call the corresponding MorphGetBufferSize to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding MorphOpenBorder function.

Use the oSrcOffset and oSrcSize parameters to control where the border control operation is applied to the source image ROI borders.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.13.3.4.1.1 Common parameters for nppiMorphOpenBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

Source image starting point relative to pSrc.

param pDst*Destination-Image Pointer.***param nDstStep***Destination-Image Line Step.***param oSizeROI***Region-Of-Interest (ROI).***param pMask**

Device memory pointer to the start address of the mask array (non-zero values select participating pixels).

param oMaskSize

Width and Height mask array.

param oAnchor

X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

param pBuffer

Pointer to device memory scratch buffer at least as large as value returned by the corresponding MorphGetBufferSize call.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes***Functions**

NppStatus **nppiMorphOpenBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer morphological open with border control.

For common parameter descriptions, see *Common parameters for nppiMorphOpenBorder functions:*.

NppStatus **nppiMorphOpenBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer morphological open with border control.

For common parameter descriptions, see *Common parameters for nppiMorphOpenBorder functions:*.

NppStatus **nppiMorphOpenBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer morphological open with border control.

For common parameter descriptions, see *Common parameters for nppiMorphOpenBorder functions*.

NppStatus **nppiMorphOpenBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer morphological open with border control.

For common parameter descriptions, see *Common parameters for nppiMorphOpenBorder functions*.

NppStatus **nppiMorphOpenBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed integer morphological open with border control.

For common parameter descriptions, see *Common parameters for nppiMorphOpenBorder functions*.

NppStatus **nppiMorphOpenBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point morphological open with border control.

For common parameter descriptions, see *Common parameters for nppiMorphOpenBorder functions*.

NppStatus **nppiMorphOpenBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point morphological open with border control.

For common parameter descriptions, see *Common parameters for nppiMorphOpenBorder functions*.

NppStatus **nppiMorphOpenBorder_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point morphological open with border control.

For common parameter descriptions, see *Common parameters for nppiMorphOpenBorder functions*:

1.13.3.5 Image Morph Top Hat Border

1.13.3.5.1 MorphToHatBorder

Source pixel minus the morphological open pixel result with border control.

Morphological top hat computes the output pixel as the source pixel minus the morphological open result of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum or minimum search.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image. The mask is centered over the source image pixel being tested.

Before calling any of the MorphTopHatBorder functions the application first needs to call the corresponding MorphGetBufferSize to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding MorphTopHatBorder function.

Use the oSrcOffset and oSrcSize parameters to control where the border control operation is applied to the source image ROI borders.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.13.3.5.1.1 Common parameters for nppiMorphTopHatBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

Source image starting point relative to pSrc.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI*Region-Of-Interest (ROI).***param pMask**

Device memory pointer to the start address of the mask array (non-zero values select participating pixels).

param oMaskSize

Width and Height mask array.

param oAnchor

X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

param pBuffer

Pointer to device memory scratch buffer at least as large as value returned by the corresponding MorphGetBufferSize call.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes*

Functions

NppStatus **nppiMorphTopHatBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer morphological top hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphTopHatBorder functions*.

NppStatus **nppiMorphTopHatBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer morphological top hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphTopHatBorder functions*.

NppStatus **nppiMorphTopHatBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer morphological top hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphTopHatBorder functions*.

NppStatus **nppiMorphTopHatBorder_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned integer morphological top hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphTopHatBorder functions*.

NppStatus **nppiMorphTopHatBorder_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed integer morphological top hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphTopHatBorder functions*.

NppStatus **nppiMorphTopHatBorder_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point morphological top hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphTopHatBorder functions*.

NppStatus **nppiMorphTopHatBorder_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point morphological top hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphTopHatBorder functions*.

```
NppStatus nppiMorphTopHatBorder_32f_C4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst,
int nDstStep, NppiSize oSizeROI, const Npp8u
*pMask, NppiSize oMaskSize, NppiPoint
oAnchor, Npp8u *pBuffer, NppiBorderType
eBorderType, NppStreamContext
nppStreamCtx)
```

4 channel 32-bit floating point morphological top hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphTopHatBorder functions*:

1.13.3.6 Image Morph Black Hat Border

1.13.3.6.1 MorphBlackHatBorder

Morphological close pixel result minus source pixel with border control.

Morphological black hat computes the output pixel as the morphological close pixel value of the pixels under the mask minus the source pixel value. Pixels whose corresponding mask values are zero do not participate in the maximum or minimum search.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image. The mask is centered over the source image pixel being tested.

Before calling any of the MorphBlackHatBorder functions the application first needs to call the corresponding MorphGetBufferSize to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding MorphBlackHatBorder function.

Use the oSrcOffset and oSrcSize parameters to control where the border control operation is applied to the source image ROI borders.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.13.3.6.1.1 Common parameters for nppiMorphBlackHatBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

Source image starting point relative to pSrc.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI*Region-Of-Interest (ROI).***param pMask**

Device memory pointer to the start address of the mask array (non-zero values select participating pixels).

param oMaskSize

Width and Height mask array.

param oAnchor

X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

param pBuffer

Pointer to device memory scratch buffer at least as large as value returned by the corresponding MorphGetBufferSize call.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes***Functions**

NppStatus **nppiMorphBlackHatBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer morphological black hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphBlackHatBorder functions*.

NppStatus **nppiMorphBlackHatBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer morphological black hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphBlackHatBorder functions*.

NppStatus **nppiMorphBlackHatBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer morphological black hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphBlackHatBorder functions*.

```
NppStatus nppiMorphBlackHatBorder_16u_C1R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16u
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp8u *pMask, NppiSize oMaskSize,
NppiPoint oAnchor, Npp8u *pBuffer,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

1 channel 16-bit unsigned integer morphological black hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphBlackHatBorder functions*.

```
NppStatus nppiMorphBlackHatBorder_16s_C1R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16s
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp8u *pMask, NppiSize oMaskSize,
NppiPoint oAnchor, Npp8u *pBuffer,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

1 channel 16-bit signed integer morphological black hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphBlackHatBorder functions*.

```
NppStatus nppiMorphBlackHatBorder_32f_C1R_Ctx( const Npp32f *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp8u *pMask, NppiSize oMaskSize,
NppiPoint oAnchor, Npp8u *pBuffer,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

1 channel 32-bit floating point morphological black hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphBlackHatBorder functions*.

```
NppStatus nppiMorphBlackHatBorder_32f_C3R_Ctx( const Npp32f *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp8u *pMask, NppiSize oMaskSize,
NppiPoint oAnchor, Npp8u *pBuffer,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

3 channel 32-bit floating point morphological black hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphBlackHatBorder functions*.


```
NppStatus nppiMorphBlackHatBorder_32f_C4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp8u *pMask, NppiSize oMaskSize,
NppiPoint oAnchor, Npp8u *pBuffer,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

4 channel 32-bit floating point morphological black hat with border control.

For common parameter descriptions, see *Common parameters for nppiMorphBlackHatBorder functions*:

1.13.3.7 Image Morph Gradient Border

1.13.3.7.1 MorphGradientBorder

Morphological dilated pixel result minus morphological eroded pixel result with border control.

Morphological gradient computes the output pixel as the morphological dilated pixel value of the pixels under the mask minus the morphological eroded pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum or minimum search.

If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image. The mask is centered over the source image pixel being tested.

Before calling any of the MorphGradientBorder functions the application first needs to call the corresponding MorphGetBufferSize to determine the amount of device memory to allocate as a working buffer. The allocated device memory is then passed as the pBuffer parameter to the corresponding MorphGradientBorder function.

Use the oSrcOffset and oSrcSize parameters to control where the border control operation is applied to the source image ROI borders.

Currently only the NPP_BORDER_REPLICATE border type operation is supported.

1.13.3.7.1.1 Common parameters for nppiMorphGradientBorder functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcSize

Source image width and height in pixels relative to pSrc.

param oSrcOffset

Source image starting point relative to pSrc.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI*Region-Of-Interest (ROI).***param pMask**

Device memory pointer to the start address of the mask array (non-zero values select participating pixels).

param oMaskSize

Width and Height mask array.

param oAnchor

X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

param pBuffer

Pointer to device memory scratch buffer at least as large as value returned by the corresponding MorphGetBufferSize call.

param eBorderType

The border type operation to be applied at source image border boundaries.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes*

Functions

NppStatus **nppiMorphGradientBorder_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned integer morphological gradient with border control.

For common parameter descriptions, see *Common parameters for nppiMorphGradientBorder functions*.

NppStatus **nppiMorphGradientBorder_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned integer morphological gradient with border control.

For common parameter descriptions, see *Common parameters for nppiMorphGradientBorder functions*.

NppStatus **nppiMorphGradientBorder_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcSize, *NppiPoint* oSrcOffset, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* *pMask, *NppiSize* oMaskSize, *NppiPoint* oAnchor, *Npp8u* *pBuffer, *NppiBorderType* eBorderType, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned integer morphological gradient with border control.

For common parameter descriptions, see *Common parameters for nppiMorphGradientBorder functions*.

```
NppStatus nppiMorphGradientBorder_16u_C1R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16u
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp8u *pMask, NppiSize oMaskSize,
NppiPoint oAnchor, Npp8u *pBuffer,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

1 channel 16-bit unsigned integer morphological gradient with border control.

For common parameter descriptions, see *Common parameters for nppiMorphGradientBorder functions*.

```
NppStatus nppiMorphGradientBorder_16s_C1R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp16s
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp8u *pMask, NppiSize oMaskSize,
NppiPoint oAnchor, Npp8u *pBuffer,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

1 channel 16-bit signed integer morphological gradient with border control.

For common parameter descriptions, see *Common parameters for nppiMorphGradientBorder functions*.

```
NppStatus nppiMorphGradientBorder_32f_C1R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp8u *pMask, NppiSize oMaskSize,
NppiPoint oAnchor, Npp8u *pBuffer,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

1 channel 32-bit floating point morphological gradient with border control.

For common parameter descriptions, see *Common parameters for nppiMorphGradientBorder functions*.

```
NppStatus nppiMorphGradientBorder_32f_C3R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize
oSrcSize, NppiPoint oSrcOffset, Npp32f
*pDst, int nDstStep, NppiSize oSizeROI,
const Npp8u *pMask, NppiSize oMaskSize,
NppiPoint oAnchor, Npp8u *pBuffer,
NppiBorderType eBorderType,
NppStreamContext nppStreamCtx)
```

3 channel 32-bit floating point morphological gradient with border control.

For common parameter descriptions, see *Common parameters for nppiMorphGradientBorder functions*.

```
NppStatus nppiMorphGradientBorder_32f_C4R_Ctx(const Npp32f *pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp8u *pMask, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u *pBuffer, NppiBorderType eBorderType, NppStreamContext nppStreamCtx)
```

4 channel 32-bit floating point morphological gradient with border control.

For common parameter descriptions, see *Common parameters for nppiMorphGradientBorder functions*.

1.14. Image Statistics Functions

Primitives for computing the statistical properties of an image.

Some statistical primitives also require scratch buffer during the computation. For details, please refer to *Scratch Buffer and Host Pointer*.

These functions can be found in the nppist library. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

1.14.1. CommonGetBufferHostSizeParameters

Common parameters for nppiGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes*.

1.14.2. Image Sum

1.14.2.1 Sum

Primitives for computing the sum of all the pixel values in an image.

Sum

Given an image $pSrc$ with width W and height H , the sum will be computed as

$$Sum = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

All the results are stored in a 64-bit double precision format, except for two primitives `nppiSum_8u64s_C1R_Ctx` and `nppiSum_8u64s_C4R_Ctx`. The sum functions require additional scratch buffer for computations.

1.14.2.2 Common parameters for nppiSum functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use `nppiSumGetBufferHostSize_XX_XXX` to determine the minimum number of bytes required.

param pSum

Device memory pointer to the computed sum.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSum_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pSum, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*

NppStatus **nppiSum_8u64s_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64s* *pSum, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image sum.

The result is 64-bit long long integer.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*

NppStatus **nppiSum_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pSum, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pSum, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pSum, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[3], *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[3], *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[3], *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[3], *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[3], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image sum ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image sum ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image sum ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[3], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image sum ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[4], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_8u64s_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64s* aSum[4], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image sum.

The result is 64-bit long long integer.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[4], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[4], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

NppStatus **nppiSum_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aSum[4], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image sum.

For common parameter descriptions, see *Common parameters for nppiSum functions include:*.

SumGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the sum primitives.

1.14.2.3 CommonSumGetBufferHostSizeParameters

Common parameters for `nppiSumGetBufferHostSize` functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: `hpBufferSize` is a *host pointer*. *Scratch Buffer and Host Pointer*.

param nppStreamCtx

Application Managed Stream Context.

return

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), *ROI Related Error Codes*.

NppStatus `nppiSumGetBufferHostSize_8u_C1R_Ctx`(*NppiSize* oSizeROI, *size_t* *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus `nppiSumGetBufferHostSize_8u64s_C1R_Ctx`(*NppiSize* oSizeROI, *size_t* *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_8u64s_C1R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus `nppiSumGetBufferHostSize_16u_C1R_Ctx`(*NppiSize* oSizeROI, *size_t* *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus `nppiSumGetBufferHostSize_16s_C1R_Ctx`(*NppiSize* oSizeROI, *size_t* *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus `nppiSumGetBufferHostSize_32f_C1R_Ctx`(*NppiSize* oSizeROI, *size_t* *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus `nppiSumGetBufferHostSize_8u_C3R_Ctx`(*NppiSize* oSizeROI, *size_t* *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_8u_AC4R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_16u_AC4R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_16s_AC4R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_8u64s_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_8u64s_C4R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

NppStatus **nppiSumGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSum_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonSumGetBufferHostSizeParameters*.

1.14.3. Image Min

1.14.3.1 Min

Primitives for computing the minimal pixel value of an image.

Min

The scratch buffer is required by the min functions.

1.14.3.2 Common parameters for nppiMin functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppiMinGetBufferHostSize_XX_XXX* to determine the minimum number of bytes required.

param pMin

Device memory pointer to the computed min.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiMin_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pMin, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus nppiMin_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* *pMin, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus nppiMin_16s_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* *pMin, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus nppiMin_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* *pMin, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus nppiMin_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* aMin[3], *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus nppiMin_16u_C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* aMin[3], *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus nppiMin_16s_C3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMin[3], *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus nppiMin_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMin[3], *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus nppiMin_8u_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* aMin[4], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus nppiMin_16u_C4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* aMin[4], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus **nppiMin_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMin[4], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus **nppiMin_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMin[4], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image min.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus **nppiMin_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* aMin[3], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image min ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus **nppiMin_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* aMin[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image min ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus **nppiMin_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMin[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image min ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

NppStatus **nppiMin_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMin[3], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image min ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMin functions include:*.

MinGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the min primitives.

1.14.3.3 CommonMinGetBufferHostSizeParameters

Common parameters for nppiMinGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes*.

NppStatus **nppiMinGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_8u_AC4R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_16u_AC4R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_16s_AC4R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

NppStatus **nppiMinGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMin_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonMinGetBufferHostSizeParameters*.

1.14.4. Image Min Index

1.14.4.1 MinIndx

Primitives for computing the minimal value and its indices (X and Y coordinates) of an image.

MinIndx

If there are several minima in the selected ROI, the function returns one on the top leftmost position.
The scratch buffer is required by the functions.

1.14.4.2 Common parameters for nppiMinIndx functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer* Use `nppiMinIndxGetBufferHostSize_XX_XXX` to determine the minimum number of bytes required.

param pMin

Device memory pointer to the computed min result.

param pIndexX

Device memory pointer to the X coordinate of the image min value.

param pIndexY

Device memory pointer to the Y coordinate of the image min value.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiMinIndx_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pMin, int *pIndexX, int *pIndexY, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image MinIndx.

For common parameter descriptions, see *Common parameters for nppiMinIndx functions include:>*.

NppStatus **nppiMinIndx_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* *pMin, int *pIndexX, int *pIndexY, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

```
NppStatus nppiMinIdx_16s_C1R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                     Npp8u *pDeviceBuffer, Npp16s *pMin, int *pIndexX, int  
                                     *pIndexY, NppStreamContext nppStreamCtx )
```

One-channel 16-bit signed image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

```
NppStatus nppiMinIdx_32f_C1R_Ctx( const Npp32f *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                     Npp8u *pDeviceBuffer, Npp32f *pMin, int *pIndexX, int  
                                     *pIndexY, NppStreamContext nppStreamCtx )
```

One-channel 32-bit floating point image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

```
NppStatus nppiMinIdx_8u_C3R_Ctx( const Npp8u *pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u  
                                     *pDeviceBuffer, Npp8u aMin[3], int aIndexX[3], int  
                                     aIndexY[3], NppStreamContext nppStreamCtx )
```

Three-channel 8-bit unsigned image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

```
NppStatus nppiMinIdx_16u_C3R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                     Npp8u *pDeviceBuffer, Npp16u aMin[3], int aIndexX[3], int  
                                     aIndexY[3], NppStreamContext nppStreamCtx )
```

Three-channel 16-bit unsigned image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

```
NppStatus nppiMinIdx_16s_C3R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                     Npp8u *pDeviceBuffer, Npp16s aMin[3], int aIndexX[3], int  
                                     aIndexY[3], NppStreamContext nppStreamCtx )
```

Three-channel 16-bit signed image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

```
NppStatus nppiMinIdx_32f_C3R_Ctx( const Npp32f *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                     Npp8u *pDeviceBuffer, Npp32f aMin[3], int aIndexX[3], int  
                                     aIndexY[3], NppStreamContext nppStreamCtx )
```

Three-channel 32-bit floating point image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

```
NppStatus nppiMinIdx_8u_C4R_Ctx( const Npp8u *pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u  
                                     *pDeviceBuffer, Npp8u aMin[4], int aIndexX[4], int  
                                     aIndexY[4], NppStreamContext nppStreamCtx )
```

Four-channel 8-bit unsigned image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

```
NppStatus nppiMinIdx_16u_C4R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                     Npp8u *pDeviceBuffer, Npp16u aMin[4], int aIndexX[4], int  
                                     aIndexY[4], NppStreamContext nppStreamCtx )
```

Four-channel 16-bit unsigned image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

NppStatus nppiMinIdx_16s_C4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMin[4], int aIndexX[4], int aIndexY[4], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

NppStatus nppiMinIdx_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMin[4], int aIndexX[4], int aIndexY[4], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image MinIdx.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

NppStatus nppiMinIdx_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* aMin[3], int aIndexX[3], int aIndexY[3], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image MinIdx ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

NppStatus nppiMinIdx_16u_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* aMin[3], int aIndexX[3], int aIndexY[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image MinIdx ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

NppStatus nppiMinIdx_16s_AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMin[3], int aIndexX[3], int aIndexY[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image MinIdx ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

NppStatus nppiMinIdx_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMin[3], int aIndexX[3], int aIndexY[3], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image MinIdx ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinIdx functions include:*.

MinIdxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MinIdx primitives.

1.14.4.3 CommonMinIndxGetBufferHostSizeParameters

Common parameters for nppiMinIndxGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiMinIndxGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_8u_C1R_Ctx.

For common parameter descriptions, see *CommonMinIndxGetBufferHostSizeParameters*.

NppStatus **nppiMinIndxGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_16u_C1R_Ctx.

For common parameter descriptions, see *CommonMinIndxGetBufferHostSizeParameters*.

NppStatus **nppiMinIndxGetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_16s_C1R_Ctx.

For common parameter descriptions, see *CommonMinIndxGetBufferHostSizeParameters*.

NppStatus **nppiMinIndxGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_32f_C1R_Ctx.

For common parameter descriptions, see *CommonMinIndxGetBufferHostSizeParameters*.

NppStatus **nppiMinIndxGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_8u_C3R_Ctx.

For common parameter descriptions, see *CommonMinIndxGetBufferHostSizeParameters*.

NppStatus **nppiMinIndxGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_16u_C3R_Ctx.

For common parameter descriptions, see *CommonMinIndxGetBufferHostSizeParameters*.

NppStatus nppiMinIdxGetBufferHostSize_16s_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_16s_C3R_Ctx.

For common parameter descriptions, see [CommonMinIdxGetBufferHostSizeParameters](#).

NppStatus nppiMinIdxGetBufferHostSize_32f_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_32f_C3R_Ctx.

For common parameter descriptions, see [CommonMinIdxGetBufferHostSizeParameters](#).

NppStatus nppiMinIdxGetBufferHostSize_8u_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_8u_C4R_Ctx.

For common parameter descriptions, see [CommonMinIdxGetBufferHostSizeParameters](#).

NppStatus nppiMinIdxGetBufferHostSize_16u_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_16u_C4R_Ctx.

For common parameter descriptions, see [CommonMinIdxGetBufferHostSizeParameters](#).

NppStatus nppiMinIdxGetBufferHostSize_16s_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_16s_C4R_Ctx.

For common parameter descriptions, see [CommonMinIdxGetBufferHostSizeParameters](#).

NppStatus nppiMinIdxGetBufferHostSize_32f_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_32f_C4R_Ctx.

For common parameter descriptions, see [CommonMinIdxGetBufferHostSizeParameters](#).

NppStatus nppiMinIdxGetBufferHostSize_8u_AC4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_8u_AC4R_Ctx.

For common parameter descriptions, see [CommonMinIdxGetBufferHostSizeParameters](#).

NppStatus nppiMinIdxGetBufferHostSize_16u_AC4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIdx_16u_AC4R_Ctx.

For common parameter descriptions, see [CommonMinIdxGetBufferHostSizeParameters](#).

NppStatus nppiMinIdxGetBufferHostSize_16s_AC4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_16u_AC4R_Ctx.

For common parameter descriptions, see [CommonMinIndxGetBufferHostSizeParameters](#).

NppStatus **nppiMinIndxGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMinIndx_32f_AC4R_Ctx.

For common parameter descriptions, see [CommonMinIndxGetBufferHostSizeParameters](#).

1.14.5. Image Max

1.14.5.1 Max

Primitives for computing the maximal pixel value of an image.

1.14.5.1.1 Common parameters for nppiMax functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use nppiMaxGetBufferHostSize_XX_XXX to determine the minimum number of bytes required.

param pMax

Device memory pointer to the computed max.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Max

The scratch buffer is required by the functions.

NppStatus nppiMax_8u_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pMax, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*

NppStatus nppiMax_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* *pMax, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*

NppStatus nppiMax_16s_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* *pMax, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*

NppStatus nppiMax_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* *pMax, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*

NppStatus nppiMax_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* aMax[3], *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*

NppStatus nppiMax_16u_C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* aMax[3], *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*

NppStatus nppiMax_16s_C3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMax[3], *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*

NppStatus nppiMax_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMax[3], *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*

NppStatus **nppiMax_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* aMax[4], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*.

NppStatus **nppiMax_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* aMax[4], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*.

NppStatus **nppiMax_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMax[4], *NppStreamContext* nppStreamCtx)

For common parameter descriptions, see *Common parameters for nppiMax functions include:*.

NppStatus **nppiMax_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMax[4], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Max.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*.

NppStatus **nppiMax_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* aMax[3], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Max ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*.

NppStatus **nppiMax_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* aMax[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Max ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*.

NppStatus **nppiMax_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMax[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Max ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*.

NppStatus **nppiMax_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMax[3], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Max ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMax functions include:*.

MaxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Max primitives.

1.14.5.2 CommonMaxGetBufferHostSizeParameters

Common parameters for nppiMaxGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiMaxGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_8u_AC4R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_16u_AC4R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_16s_AC4R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

NppStatus **nppiMaxGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMax_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonMaxGetBufferHostSizeParameters*.

1.14.6. Image Max Index

1.14.6.1 MaxIdx

Primitives for computing the maximal value and its indices (X and Y coordinates) of an image.

MaxIdx

If there are several maxima in the selected region of interest, the function returns one on the top leftmost position.

The scratch buffer is required by the functions.

1.14.6.2 Common parameters for nppiMaxIdx functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer* Use nppiMaxIdxGetBufferHostSize_XX_XXX to determine the minimum number of bytes required.

param pMax

Device memory pointer to the computed max result.

param pIndexX

Device memory pointer to the X coordinate of the image max value.

param pIndexY

Device memory pointer to the Y coordinate of the image max value.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiMaxIdx_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* *pMax, int *pIndexX, int *pIndexY, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*

NppStatus **nppiMaxIdx_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* *pMax, int *pIndexX, int *pIndexY, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus **nppiMaxIdx_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* *pMax, int *pIndexX, int *pIndexY, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus **nppiMaxIdx_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* *pMax, int *pIndexX, int *pIndexY, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus **nppiMaxIdx_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* aMax[3], int aIndexX[3], int aIndexY[3], *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus **nppiMaxIdx_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* aMax[3], int aIndexX[3], int aIndexY[3], *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus **nppiMaxIdx_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMax[3], int aIndexX[3], int aIndexY[3], *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus **nppiMaxIdx_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMax[3], int aIndexX[3], int aIndexY[3], *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus **nppiMaxIdx_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* aMax[4], int aIndexX[4], int aIndexY[4], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus **nppiMaxIdx_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* aMax[4], int aIndexX[4], int aIndexY[4], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus nppiMaxIdx_16s_C4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMax[4], int alIndexX[4], int alIndexY[4], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus nppiMaxIdx_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMax[4], int alIndexX[4], int alIndexY[4], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image MaxIdx.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus nppiMaxIdx_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp8u* aMax[3], int alIndexX[3], int alIndexY[3], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image MaxIdx ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus nppiMaxIdx_16u_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16u* aMax[3], int alIndexX[3], int alIndexY[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image MaxIdx ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus nppiMaxIdx_16s_AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp16s* aMax[3], int alIndexX[3], int alIndexY[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image MaxIdx ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

NppStatus nppiMaxIdx_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp32f* aMax[3], int alIndexX[3], int alIndexY[3], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image MaxIdx ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMaxIdx functions include:*.

MaxIdxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MaxIdx primitives.

1.14.6.3 CommonMaxIndxGetBufferHostSizeParameters

Common parameters for nppiMaxIndxGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiMaxIndxGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_C1R_Ctx.

For common parameter descriptions, see *CommonMaxIndxGetBufferHostSizeParameters.*

NppStatus **nppiMaxIndxGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_C1R_Ctx.

For common parameter descriptions, see *CommonMaxIndxGetBufferHostSizeParameters.*

NppStatus **nppiMaxIndxGetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16s_C1R_Ctx.

For common parameter descriptions, see *CommonMaxIndxGetBufferHostSizeParameters.*

NppStatus **nppiMaxIndxGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_32f_C1R_Ctx.

For common parameter descriptions, see *CommonMaxIndxGetBufferHostSizeParameters.*

NppStatus **nppiMaxIndxGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_8u_C3R_Ctx.

For common parameter descriptions, see *CommonMaxIndxGetBufferHostSizeParameters.*

NppStatus **nppiMaxIndxGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx_16u_C3R_Ctx.

For common parameter descriptions, see *CommonMaxIndxGetBufferHostSizeParameters.*

NppStatus nppiMaxIdxGetBufferHostSize_16s_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16s_C3R_Ctx.

For common parameter descriptions, see [CommonMaxIdxGetBufferHostSizeParameters](#).

NppStatus nppiMaxIdxGetBufferHostSize_32f_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIdx_32f_C3R_Ctx.

For common parameter descriptions, see [CommonMaxIdxGetBufferHostSizeParameters](#).

NppStatus nppiMaxIdxGetBufferHostSize_8u_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIdx_8u_C4R_Ctx.

For common parameter descriptions, see [CommonMaxIdxGetBufferHostSizeParameters](#).

NppStatus nppiMaxIdxGetBufferHostSize_16u_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16u_C4R_Ctx.

For common parameter descriptions, see [CommonMaxIdxGetBufferHostSizeParameters](#).

NppStatus nppiMaxIdxGetBufferHostSize_16s_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16s_C4R_Ctx.

For common parameter descriptions, see [CommonMaxIdxGetBufferHostSizeParameters](#).

NppStatus nppiMaxIdxGetBufferHostSize_32f_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIdx_32f_C4R_Ctx.

For common parameter descriptions, see [CommonMaxIdxGetBufferHostSizeParameters](#).

NppStatus nppiMaxIdxGetBufferHostSize_8u_AC4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIdx_8u_AC4R_Ctx.

For common parameter descriptions, see [CommonMaxIdxGetBufferHostSizeParameters](#).

NppStatus nppiMaxIdxGetBufferHostSize_16u_AC4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16u_AC4R_Ctx.

For common parameter descriptions, see [CommonMaxIdxGetBufferHostSizeParameters](#).

NppStatus nppiMaxIdxGetBufferHostSize_16s_AC4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIdx_16u_AC4R_Ctx.

For common parameter descriptions, see [CommonMaxIdxGetBufferHostSizeParameters](#).

NppStatus **nppiMaxIdxGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiMaxIdx_32f_AC4R_Ctx.

For common parameter descriptions, see [CommonMaxIdxGetBufferHostSizeParameters](#).

1.14.7. Image MinMax

1.14.7.1 MinMax

Primitives for computing both the minimal and the maximal values of an image.

MinMax

The functions require the device scratch buffer.

1.14.7.2 Common parameters for nppiMinMax functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pMin

Device memory pointer to the computed minimal result.

param pMax

Device memory pointer to the computed maximal result.

param pDeviceBuffer

Buffer to a scratch memory. Use nppiMinMax_XX_XXX to determine the minimum number of bytes required.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiMinMax_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u*
*pMin, *Npp8u* *pMax, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 8-bit unsigned image MinMax.

NppStatus **nppiMinMax_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp16u* *pMin, *Npp16u* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

NppStatus **nppiMinMax_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp16s* *pMin, *Npp16s* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

NppStatus **nppiMinMax_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32f* *pMin, *Npp32f* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

NppStatus **nppiMinMax_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* aMin[3], *Npp8u* aMax[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

NppStatus **nppiMinMax_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp16u* aMin[3], *Npp16u* aMax[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

NppStatus **nppiMinMax_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp16s* aMin[3], *Npp16s* aMax[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

NppStatus **nppiMinMax_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32f* aMin[3], *Npp32f* aMax[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

NppStatus **nppiMinMax_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* aMin[3], *Npp8u* aMax[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image MinMax ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

NppStatus **nppiMinMax_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp16u* aMin[3], *Npp16u* aMax[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image MinMax ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

```
NppStatus nppiMinMax_16s_AC4R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                     Npp16s aMin[3], Npp16s aMax[3], Npp8u *pDeviceBuffer,  
                                     NppStreamContext nppStreamCtx )
```

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

```
NppStatus nppiMinMax_32f_AC4R_Ctx( const Npp32f *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                     Npp32f aMin[3], Npp32f aMax[3], Npp8u *pDeviceBuffer,  
                                     NppStreamContext nppStreamCtx )
```

Four-channel 32-bit floating point image MinMax ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

```
NppStatus nppiMinMax_8u_C4R_Ctx( const Npp8u *pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u  
                                   aMin[4], Npp8u aMax[4], Npp8u *pDeviceBuffer,  
                                   NppStreamContext nppStreamCtx )
```

Four-channel 8-bit unsigned image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

```
NppStatus nppiMinMax_16u_C4R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                   Npp16u aMin[4], Npp16u aMax[4], Npp8u *pDeviceBuffer,  
                                   NppStreamContext nppStreamCtx )
```

Four-channel 16-bit unsigned image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

```
NppStatus nppiMinMax_16s_C4R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                   Npp16s aMin[4], Npp16s aMax[4], Npp8u *pDeviceBuffer,  
                                   NppStreamContext nppStreamCtx )
```

Four-channel 16-bit signed image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

```
NppStatus nppiMinMax_32f_C4R_Ctx( const Npp32f *pSrc, int nSrcStep, NppiSize oSizeROI,  
                                   Npp32f aMin[4], Npp32f aMax[4], Npp8u *pDeviceBuffer,  
                                   NppStreamContext nppStreamCtx )
```

Four-channel 32-bit floating point image MinMax.

For common parameter descriptions, see *Common parameters for nppiMinMax functions include:*.

MinMaxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MinMax primitives.

1.14.7.3 CommonMinMaxGetBufferHostSizeParameters

Common parameters for nppiMinMaxGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus nppiMinMaxGetBufferHostSize_8u_C1R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMinMax_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus nppiMinMaxGetBufferHostSize_16u_C1R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMinMax_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus nppiMinMaxGetBufferHostSize_16s_C1R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMinMax_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus nppiMinMaxGetBufferHostSize_32f_C1R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMinMax_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus nppiMinMaxGetBufferHostSize_8u_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMinMax_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus nppiMinMaxGetBufferHostSize_16u_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMinMax_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus nppiMinMaxGetBufferHostSize_16s_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMinMax_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus **nppiMinMaxGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiMinMax_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus **nppiMinMaxGetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiMinMax_8u_AC4R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus **nppiMinMaxGetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiMinMax_16u_AC4R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus **nppiMinMaxGetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiMinMax_16s_AC4R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus **nppiMinMaxGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiMinMax_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus **nppiMinMaxGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Buffer size for *nppiMinMax_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus **nppiMinMaxGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiMinMax_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus **nppiMinMaxGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiMinMax_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

NppStatus **nppiMinMaxGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

For common parameter descriptions, see *CommonMinMaxGetBufferHostSizeParameters*.

1.14.8. Image Mean

1.14.8.1 Mean

Primitives for computing the arithmetic mean of all the pixel values in an image.

Mean

Given an image *pSrc* with width *W* and height *H*, the arithmetic mean will be computed as

$$Mean = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

The mean functions require additional scratch buffer for computations.

1.14.8.2 Common parameters for nppiMean functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pMask

Mask-Image Pointer.

param nMaskStep

Mask-Image Line Step.

param nCOI

Channel_of_Interest Number.

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer* Use *nppiMeanGetBufferHostSize_XX_XXX* to determine the minimum number of bytes required.

param pMean

Device memory pointer to the computed mean result.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, *ROI Related Error Codes*, or `NPP_COI_ERROR` if an invalid channel of interest is specified.

NppStatus **nppiMean_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus **nppiMean_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus **nppiMean_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus **nppiMean_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus **nppiMean_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[3], *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus **nppiMean_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[3], *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus **nppiMean_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[3], *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus **nppiMean_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[3], *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus nppiMean_8u_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[4], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus nppiMean_16u_C4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[4], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus nppiMean_16s_C4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[4], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus nppiMean_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[4], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus nppiMean_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[3], *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Mean ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus nppiMean_16u_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Mean ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus nppiMean_16s_AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[3], *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Mean ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus nppiMean_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* aMean[3], *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Mean ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

NppStatus nppiMean_8u_C1MR_Ctx(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit unsigned image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

```
NppStatus nppiMean_8s_C1MR_Ctx( const Npp8s *pSrc, int nSrcStep, const Npp8u *pMask, int  
                                nMaskStep, NppiSize oSizeROI, Npp8u *pDeviceBuffer, Npp64f  
                                *pMean, NppStreamContext nppStreamCtx )
```

Masked one-channel 8-bit signed image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

```
NppStatus nppiMean_16u_C1MR_Ctx( const Npp16u *pSrc, int nSrcStep, const Npp8u *pMask, int  
                                nMaskStep, NppiSize oSizeROI, Npp8u *pDeviceBuffer,  
                                Npp64f *pMean, NppStreamContext nppStreamCtx )
```

Masked one-channel 16-bit unsigned image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

```
NppStatus nppiMean_32f_C1MR_Ctx( const Npp32f *pSrc, int nSrcStep, const Npp8u *pMask, int  
                                nMaskStep, NppiSize oSizeROI, Npp8u *pDeviceBuffer,  
                                Npp64f *pMean, NppStreamContext nppStreamCtx )
```

Masked one-channel 32-bit floating point image Mean.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

```
NppStatus nppiMean_8u_C3CMR_Ctx( const Npp8u *pSrc, int nSrcStep, const Npp8u *pMask, int  
                                nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u  
                                *pDeviceBuffer, Npp64f *pMean, NppStreamContext  
                                nppStreamCtx )
```

Masked three-channel 8-bit unsigned image Mean affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

```
NppStatus nppiMean_8s_C3CMR_Ctx( const Npp8s *pSrc, int nSrcStep, const Npp8u *pMask, int  
                                nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u  
                                *pDeviceBuffer, Npp64f *pMean, NppStreamContext  
                                nppStreamCtx )
```

Masked three-channel 8-bit signed image Mean affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

```
NppStatus nppiMean_16u_C3CMR_Ctx( const Npp16u *pSrc, int nSrcStep, const Npp8u *pMask, int  
                                nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u  
                                *pDeviceBuffer, Npp64f *pMean, NppStreamContext  
                                nppStreamCtx )
```

Masked three-channel 16-bit unsigned image Mean affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

```
NppStatus nppiMean_32f_C3CMR_Ctx( const Npp32f *pSrc, int nSrcStep, const Npp8u *pMask, int  
                                nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u  
                                *pDeviceBuffer, Npp64f *pMean, NppStreamContext  
                                nppStreamCtx )
```

Masked three-channel 32-bit floating point image Mean affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiMean functions include:*.

MeanGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean primitives.

1.14.8.3 CommonMeanGetBufferHostSizeParameters

Common parameters for nppiMeanGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiMeanGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_8u_C1R_Ctx*.

NppStatus **nppiMeanGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_8u_AC4R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_16u_AC4R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_16s_AC4R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_8u_C1MR_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_8s_C1MR_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_16u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_16u_C1MR_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_32f_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_32f_C1MR_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_8u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_8u_C3CMR_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_8s_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_8s_C3CMR_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_16u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_16u_C3CMR_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

NppStatus **nppiMeanGetBufferHostSize_32f_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_32f_C3CMR_Ctx*.

For common parameter descriptions, see *CommonMeanGetBufferHostSizeParameters*.

1.14.9. Image Mean StdDev

1.14.9.1 Mean_StdDev

Primitives for computing both the arithmetic mean and the standard deviation of an image.

Mean_StdDev

Given an image $pSrc$ with width W and height H , the mean and the standard deviation will be computed as

$$Mean = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$
$$StdDev = \sqrt{\frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} (pSrc(j, i) - Mean)^2}$$

The Mean_StdDev primitives require additional scratch buffer for computations.

1.14.9.2 Common parameters for nppiMean_StdDev functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pMask

Mask-Image Pointer.

param nMaskStep

Mask-Image Line Step.

param nCOI

Channel_of_Interest Number.

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer* Use *MeanStdDevGetBufferHostSize* to determine the minimum number of bytes required.

param pMean

Device memory pointer to the computed mean.

param pStdDev

Device memory pointer to the computed standard deviation.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, *ROI Related Error Codes*, or NPP_COI_ERROR if an invalid channel of interest is specified.

NppStatus **nppiMean_StdDev_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

NppStatus **nppiMean_StdDev_8s_C1R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

NppStatus **nppiMean_StdDev_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

NppStatus **nppiMean_StdDev_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

NppStatus **nppiMean_StdDev_8u_C1MR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit unsigned image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

NppStatus **nppiMean_StdDev_8s_C1MR_Ctx**(const *Npp8s* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit signed image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

NppStatus **nppiMean_StdDev_16u_C1MR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

Masked one-channel 16-bit unsigned image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

```
NppStatus nppiMean_StdDev_32f_C1MR_Ctx( const Npp32f *pSrc, int nSrcStep, const Npp8u
                                         *pMask, int nMaskStep, NppiSize oSizeROI, Npp8u
                                         *pDeviceBuffer, Npp64f *pMean, Npp64f *pStdDev,
                                         NppStreamContext nppStreamCtx)
```

Masked one-channel 32-bit floating point image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

Channel Mean_StdDev

See *Channel-of-Interest API*.

```
NppStatus nppiMean_StdDev_8u_C3CR_Ctx( const Npp8u *pSrc, int nSrcStep, NppiSize oSizeROI,
                                         int nCOI, Npp8u *pDeviceBuffer, Npp64f *pMean,
                                         Npp64f *pStdDev, NppStreamContext
                                         nppStreamCtx)
```

Three-channel 8-bit unsigned image Mean_StdDev affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

```
NppStatus nppiMean_StdDev_8s_C3CR_Ctx( const Npp8s *pSrc, int nSrcStep, NppiSize oSizeROI,
                                         int nCOI, Npp8u *pDeviceBuffer, Npp64f *pMean,
                                         Npp64f *pStdDev, NppStreamContext
                                         nppStreamCtx)
```

Three-channel 8-bit signed image Mean_StdDev affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

```
NppStatus nppiMean_StdDev_16u_C3CR_Ctx( const Npp16u *pSrc, int nSrcStep, NppiSize
                                         oSizeROI, int nCOI, Npp8u *pDeviceBuffer, Npp64f
                                         *pMean, Npp64f *pStdDev, NppStreamContext
                                         nppStreamCtx)
```

Three-channel 16-bit unsigned image Mean_StdDev affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

```
NppStatus nppiMean_StdDev_32f_C3CR_Ctx( const Npp32f *pSrc, int nSrcStep, NppiSize
                                         oSizeROI, int nCOI, Npp8u *pDeviceBuffer, Npp64f
                                         *pMean, Npp64f *pStdDev, NppStreamContext
                                         nppStreamCtx)
```

Three-channel 32-bit floating point image Mean_StdDev affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

NppStatus **nppiMean_StdDev_8u_C3CMR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit unsigned image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

NppStatus **nppiMean_StdDev_8s_C3CMR_Ctx**(const *Npp8s* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit signed image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

NppStatus **nppiMean_StdDev_16u_C3CMR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

Masked three-channel 16-bit unsigned image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

NppStatus **nppiMean_StdDev_32f_C3CMR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp8u* *pDeviceBuffer, *Npp64f* *pMean, *Npp64f* *pStdDev, *NppStreamContext* nppStreamCtx)

Masked three-channel 32-bit floating point image Mean_StdDev.

For common parameter descriptions, see *Common parameters for nppiMean_StdDev functions include:*.

Unnamed Group

NppStatus **nppiMeanStdDevGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_StdDev_8u_C1R_Ctx*.

1.14.9.3 MeanStdDevGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean_StdDev primitives.

1.14.9.4 Common parameters for MeanStdDevGetBufferHostSize functions include:

NppStatus **nppiMeanStdDevGetBufferHostSize_8s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_StdDev_8s_C1R_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

NppStatus **nppiMeanStdDevGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_StdDev_16u_C1R_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

NppStatus **nppiMeanStdDevGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_StdDev_32f_C1R_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

NppStatus **nppiMeanStdDevGetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_StdDev_8u_C1MR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

NppStatus **nppiMeanStdDevGetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_StdDev_8s_C1MR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

```
NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1MR_Ctx(NppiSize oSizeROI, size_t
                                                    *hpBufferSize,
                                                    NppStreamContext
                                                    nppStreamCtx)
```

Buffer size for *nppiMean_StdDev_16u_C1MR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

```
NppStatus nppiMeanStdDevGetBufferHostSize_32f_C1MR_Ctx(NppiSize oSizeROI, size_t
                                                    *hpBufferSize,
                                                    NppStreamContext
                                                    nppStreamCtx)
```

Buffer size for *nppiMean_StdDev_32f_C1MR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

```
NppStatus nppiMeanStdDevGetBufferHostSize_8u_C3CR_Ctx(NppiSize oSizeROI, size_t
                                                    *hpBufferSize,
                                                    NppStreamContext
                                                    nppStreamCtx)
```

Buffer size for *nppiMean_StdDev_8u_C3CR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

```
NppStatus nppiMeanStdDevGetBufferHostSize_8s_C3CR_Ctx(NppiSize oSizeROI, size_t
                                                    *hpBufferSize,
                                                    NppStreamContext
                                                    nppStreamCtx)
```

Buffer size for *nppiMean_StdDev_8s_C3CR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

```
NppStatus nppiMeanStdDevGetBufferHostSize_16u_C3CR_Ctx(NppiSize oSizeROI, size_t
                                                    *hpBufferSize,
                                                    NppStreamContext
                                                    nppStreamCtx)
```

Buffer size for *nppiMean_StdDev_16u_C3CR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

```
NppStatus nppiMeanStdDevGetBufferHostSize_32f_C3CR_Ctx(NppiSize oSizeROI, size_t
                                                    *hpBufferSize,
                                                    NppStreamContext
                                                    nppStreamCtx)
```

Buffer size for *nppiMean_StdDev_32f_C3CR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

NppStatus **nppiMeanStdDevGetBufferHostSize_8u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_StdDev_8u_C3CMR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

NppStatus **nppiMeanStdDevGetBufferHostSize_8s_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_StdDev_8s_C3CMR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

NppStatus **nppiMeanStdDevGetBufferHostSize_16u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_StdDev_16u_C3CMR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

NppStatus **nppiMeanStdDevGetBufferHostSize_32f_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMean_StdDev_32f_C3CMR_Ctx*.

For common parameter descriptions, see *Common parameters for MeanStdDevGetBufferHostSize functions include:*.

1.14.10. Image Norms

1.14.10.1 Image Norms

Primitives for computing the norms of an image, the norms of difference, and the relative errors of two images. Given an image *pSrc* with width *W* and height *H*,

1. The infinity norm (Norm_Inf) is defined as the largest absolute pixel value of the image.
2. The L1 norm (Norm_L1) is defined as the sum of the absolute pixel value of the image, i.e.,

$$Norm\_L1 = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc(j, i)|$$

3. The L2 norm (Norm_L2) is defined as the square root of the sum of the squared absolute pixel value of the image, i.e.,

$$Norm\_L2 = \sqrt{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc(j, i)|^2}$$

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*,

1. The infinity norm of difference (NormDiff_Inf) is defined as the largest absolute difference between pixels of two images.
2. The L1 norm of difference (NormDiff_L1) is defined as the sum of the absolute difference between pixels of two images, i.e.,

$$NormDiff\_L1 = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|$$

3. The L2 norm of difference (NormDiff_L2) is defined as the squared root of the sum of the squared absolute difference between pixels of two images, i.e.,

$$NormDiff\_L2 = \sqrt{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|^2}$$

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*,

1. The relative error for the infinity norm of difference (NormRel_Inf) is defined as NormDiff_Inf divided by the infinity norm of the second image, i.e.,

$$NormRel\_Inf = \frac{NormDiff\_Inf}{Norm\_Inf_{src2}}$$

2. The relative error for the L1 norm of difference (NormRel_L1) is defined as NormDiff_L1 divided by the L1 norm of the second image, i.e.,

$$NormRel\_L1 = \frac{NormDiff\_L1}{Norm\_L1_{src2}}$$

3. The relative error for the L2 norm of difference (NormRel_L2) is defined as NormDiff_L2 divided by the L2 norm of the second image, i.e.,

$$NormRel\_L2 = \frac{NormDiff\_L2}{Norm\_L2_{src2}}$$

The norm functions require the addition device scratch buffer for the computations.

1.14.10.1.1 Common parameters for nppiNorm functions include:

param pSrc1
Source-Image Pointer.

param nSrc1Step

Source-Image Line Step.

param pSrc2

Source-Image Pointer.

param nSrc2Step

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pMask

Mask-Image Pointer.

param nMaskStep

Mask-Image Line Step.

param nCOI

Channel_of_Interest Number.

param pNorm

Device memory pointer to the norm value.

param pNormDiff

Device memory pointer to the computed norm of differences.

param pNormRel

Device memory pointer to the computed relative error for the infinity norm of two images.

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use `nppiNormInfGetBufferHostSize_XX_XXX` to compute the required size (in bytes).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, *ROI Related Error Codes*, or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

1.14.10.2 Image Norm Inf

1.14.10.2.1 Norm_Inf

Primitives for computing the infinity norm of an image.

Basic Norm_Inf

NppStatus nppiNorm_Inf_8u_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI,
Npp64f *pNorm, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 8-bit unsigned image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNorm_Inf_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI,
Npp64f *pNorm, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 16-bit unsigned image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNorm_Inf_16s_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI,
Npp64f *pNorm, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 16-bit signed image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNorm_Inf_32s_C1R_Ctx(const *Npp32s* *pSrc, int nSrcStep, *NppiSize* oSizeROI,
Npp64f *pNorm, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 32-bit signed image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNorm_Inf_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI,
Npp64f *pNorm, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 32-bit floating point image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNorm_Inf_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI,
Npp64f aNorm[3], *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

Three-channel 8-bit unsigned image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNorm_Inf_16u_C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI,
Npp64f aNorm[3], *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

Three-channel 16-bit unsigned image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNorm_Inf_16s_C3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI,
Npp64f aNorm[3], *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

Three-channel 16-bit signed image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_Inf_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_Inf_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Norm_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_Inf_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Norm_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_Inf_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Norm_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_Inf_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Norm_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_Inf_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_Inf_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_Inf_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_Inf_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_Inf_8u_C1MR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit unsigned image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_Inf_8s_C1MR_Ctx**(const *Npp8s* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit signed image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_Inf_16u_C1MR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 16-bit unsigned image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_Inf_32f_C1MR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 32-bit floating point image Norm_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_Inf_8u_C3CMR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit unsigned image Norm_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_Inf_8s_C3CMR_Ctx**(const *Npp8s* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit signed image Norm_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_Inf_16u_C3CMR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 16-bit unsigned image Norm_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_Inf_32f_C3CMR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 32-bit floating point image Norm_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NormInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm_Inf primitives.

1.14.10.2.2 CommonNormInfGetBufferHostSizeParameters

Common parameters for nppiNormInfGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiNormInfGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_Inf_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters.*

NppStatus **nppiNormInfGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_Inf_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters.*

NppStatus **nppiNormInfGetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_Inf_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters.*

NppStatus **nppiNormInfGetBufferHostSize_32s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_Inf_32s_C1R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters.*

NppStatus **nppiNormInfGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_8u_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_8s_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_16u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_16u_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_32f_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_32f_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_8u_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_16u_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_16s_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus **nppiNormInfGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_Inf_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus nppiNormInfGetBufferHostSize_8u_C3CMR_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_Inf_8u_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus nppiNormInfGetBufferHostSize_8s_C3CMR_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_Inf_8s_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus nppiNormInfGetBufferHostSize_16u_C3CMR_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_Inf_16u_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

NppStatus nppiNormInfGetBufferHostSize_32f_C3CMR_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_Inf_32f_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormInfGetBufferHostSizeParameters*.

1.14.10.3 Image Norm L1

1.14.10.3.1 Norm_L1

Primitives for computing the L1 norm of an image.

Basic Norm_L1

NppStatus nppiNorm_L1_8u_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNorm_L1_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNorm_L1_16s_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L1_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L1_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L1_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L1_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L1_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L1_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Norm_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L1_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Norm_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L1_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Norm_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L1_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Norm_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_8u_C1MR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit unsigned image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_8s_C1MR_Ctx**(const *Npp8s* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit signed image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_16u_C1MR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 16-bit unsigned image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_32f_C1MR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 32-bit floating point image Norm_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_8u_C3CMR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit unsigned image Norm_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_8s_C3CMR_Ctx**(const *Npp8s* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit signed image Norm_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_16u_C3CMR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 16-bit unsigned image Norm_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNorm_L1_32f_C3CMR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 32-bit floating point image Norm_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NormL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm_L1 primitives.

1.14.10.3.2 CommonNormL1GetBufferHostSizeParameters

Common parameters for nppiNormL1GetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes*.

NppStatus **nppiNormL1GetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_8u_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_8s_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_16u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_16u_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_32f_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_32f_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_8u_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_16u_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_16s_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L1_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L1_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L1_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L1_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_8u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L1_8u_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_8s_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L1_8s_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_16u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L1_16u_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

NppStatus **nppiNormL1GetBufferHostSize_32f_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L1_32f_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormL1GetBufferHostSizeParameters*.

1.14.10.4 Image Norm L2

1.14.10.4.1 Norm_L2

Primitives for computing the L2 norm of an image.

Basic Norm_L2

Computes the L2 norm of an image.

NppStatus **nppiNorm_L2_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Norm_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Norm_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_16s_AC4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Norm_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Norm_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp64f* aNorm[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_8u_C1MR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit unsigned image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_8s_C1MR_Ctx**(const *Npp8s* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit signed image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_16u_C1MR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 16-bit unsigned image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_32f_C1MR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 32-bit floating point image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_8u_C3CMR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit unsigned image Norm_L2.

NppStatus **nppiNorm_L2_8s_C3CMR_Ctx**(const *Npp8s* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit signed image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_16u_C3CMR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 16-bit unsigned image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNorm_L2_32f_C3CMR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 32-bit floating point image Norm_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NormL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm_L2 primitives.

1.14.10.4.2 CommonNormL2GetBufferHostSizeParameters

Common parameters for nppiNormL2GetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiNormL2GetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_8u_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_8s_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_16u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_16u_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_32f_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_32f_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Buffer size for *nppiNorm_L2_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_8u_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_16u_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_16s_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_8u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNorm_L2_8u_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_8s_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_8s_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_16u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_16u_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

NppStatus **nppiNormL2GetBufferHostSize_32f_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for *nppiNorm_L2_32f_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormL2GetBufferHostSizeParameters*.

1.14.10.5 Image NormDiff_Inf

1.14.10.5.1 NormDiff_Inf

Primitives for computing the infinity norm of difference of pixels between two images.

Basic NormDiff_Inf

NppStatus **nppiNormDiff_Inf_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u*
*pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f*
*pNormDiff, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 8-bit unsigned image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormDiff_Inf_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u*
*pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f*
*pNormDiff, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 16-bit unsigned image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormDiff_Inf_16s_C1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s*
*pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f*
*pNormDiff, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 16-bit signed image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormDiff_Inf_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormDiff_Inf_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormDiff_Inf_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormDiff_Inf_16s_C3R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormDiff_Inf_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image NormDiff_Inf.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **aNormDiff** – Array that contains computed Inf-norm of differences.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, *ROI Related Error Codes*, or NPP_NOT_EVEN_STEP_ERROR if an invalid floating-point image is specified.

NppStatus **nppiNormDiff_Inf_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image NormDiff_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image NormDiff_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_16s_AC4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image NormDiff_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image NormDiff_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_16s_C4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_8u_C1MR_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit unsigned images NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_8s_C1MR_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit signed images NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_16u_C1MR_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 16-bit unsigned images NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_32f_C1MR_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 32-bit floating point images NormDiff_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_8u_C3CMR_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit unsigned image NormDiff_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_8s_C3CMR_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit signed image NormDiff_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_Inf_16u_C3CMR_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 16-bit unsigned image NormDiff_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormDiff_Inf_32f_C3CMR_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 32-bit floating point image NormDiff_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

1.14.10.5.2 CommonNormDiffInfGetBufferHostSizeParameters

Common parameters for nppiNormDiffInfGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiNormDiffInfGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_8u_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_8s_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_16u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_16u_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_32f_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_32f_C1MR_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_8u_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_16u_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_16s_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_8u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_8u_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_8s_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_8s_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_16u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_16u_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

NppStatus **nppiNormDiffInfGetBufferHostSize_32f_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiNormDiff_Inf_32f_C3CMR_Ctx*.

For common parameter descriptions, see *CommonNormDiffInfGetBufferHostSizeParameters*.

1.14.10.6 Image NormDiff L1

1.14.10.6.1 NormDiff_L1

Primitives for computing the L1 norm of difference of pixels between two images.

Basic NormDiff_L1

NppStatus **nppiNormDiff_L1_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_16s_C1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_16s_C3R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_32f_C3R_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_8u_AC4R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image NormDiff_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_16u_AC4R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image NormDiff_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_16s_AC4R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image NormDiff_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_32f_AC4R_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image NormDiff_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_8u_C4R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_16u_C4R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_16s_C4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_8u_C1MR_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit unsigned image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_8s_C1MR_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit signed image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_16u_C1MR_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 16-bit unsigned image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L1_32f_C1MR_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 32-bit floating point image NormDiff_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_8u_C3CMR_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit unsigned image NormDiff_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_8s_C3CMR_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit signed image NormDiff_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_16u_C3CMR_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 16-bit unsigned image NormDiff_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L1_32f_C3CMR_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 32-bit floating point image NormDiff_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NormDiffL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_L1 primitives.

1.14.10.6.2 CommonNormDiffL1GetBufferHostSizeParameters

Common parameters for nppiNormDiffL1GetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes*.

NppStatus **nppiNormDiffL1GetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C1R_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C1R_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C1R_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C1R_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C1MR_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8s_C1MR_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_16u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C1MR_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_32f_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C1MR_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C3R_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C3R_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16s_C3R_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C3R_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_8u_C4R_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C4R_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL1GetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_16s_C4R_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL1GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL1GetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_32f_C4R_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL1GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL1GetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_8u_AC4R_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL1GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL1GetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_16u_AC4R_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL1GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL1GetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_16s_AC4R_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL1GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL1GetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_32f_AC4R_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL1GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL1GetBufferHostSize_8u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_8u_C3CMR_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL1GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL1GetBufferHostSize_8s_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L1_8s_C3CMR_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL1GetBufferHostSizeParameters*](#).

NppStatus nppiNormDiffL1GetBufferHostSize_16u_C3CMR_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_16u_C3CMR_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

NppStatus nppiNormDiffL1GetBufferHostSize_32f_C3CMR_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L1_32f_C3CMR_Ctx.

For common parameter descriptions, see *CommonNormDiffL1GetBufferHostSizeParameters*.

1.14.10.7 Image NormDiff L2

1.14.10.7.1 NormDiff_L2

Primitives for computing the L2 norm of difference of pixels between two images.

Basic NormDiff_L2

NppStatus nppiNormDiff_L2_8u_C1R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L2_16u_C1R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L2_16s_C1R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormDiff_L2_32f_C1R_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_16s_C3R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image NormDiff_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image NormDiff_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_16s_AC4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image NormDiff_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image NormDiff_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_16s_C4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormDiff[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_8u_C1MR_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit unsigned image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_8s_C1MR_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit signed image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormDiff_L2_16u_C1MR_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormDiff, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 16-bit unsigned image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormDiff_L2_32f_C1MR_Ctx( const Npp32f *pSrc1, int nSrc1Step, const Npp32f
                                         *pSrc2, int nSrc2Step, const Npp8u *pMask, int
                                         nMaskStep, NppiSize oSizeROI, Npp64f *pNormDiff,
                                         Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx )
```

Masked one-channel 32-bit floating point image NormDiff_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormDiff_L2_8u_C3CMR_Ctx( const Npp8u *pSrc1, int nSrc1Step, const Npp8u
                                         *pSrc2, int nSrc2Step, const Npp8u *pMask, int
                                         nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f
                                         *pNormDiff, Npp8u *pDeviceBuffer,
                                         NppStreamContext nppStreamCtx )
```

Masked three-channel 8-bit unsigned image NormDiff_L2 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormDiff_L2_8s_C3CMR_Ctx( const Npp8s *pSrc1, int nSrc1Step, const Npp8s
                                         *pSrc2, int nSrc2Step, const Npp8u *pMask, int
                                         nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f
                                         *pNormDiff, Npp8u *pDeviceBuffer,
                                         NppStreamContext nppStreamCtx )
```

Masked three-channel 8-bit signed image NormDiff_L2 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormDiff_L2_16u_C3CMR_Ctx( const Npp16u *pSrc1, int nSrc1Step, const Npp16u
                                         *pSrc2, int nSrc2Step, const Npp8u *pMask, int
                                         nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f
                                         *pNormDiff, Npp8u *pDeviceBuffer,
                                         NppStreamContext nppStreamCtx )
```

Masked three-channel 16-bit unsigned image NormDiff_L2 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormDiff_L2_32f_C3CMR_Ctx( const Npp32f *pSrc1, int nSrc1Step, const Npp32f
                                         *pSrc2, int nSrc2Step, const Npp8u *pMask, int
                                         nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f
                                         *pNormDiff, Npp8u *pDeviceBuffer,
                                         NppStreamContext nppStreamCtx )
```

Masked three-channel 32-bit floating point image NormDiff_L2 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NormDiffL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_L2 primitives.

1.14.10.7.2 CommonNormDiffL2GetBufferHostSizeParameters

Common parameters for nppiNormDiffL2GetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiNormDiffL2GetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C1R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters.*

NppStatus **nppiNormDiffL2GetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C1R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters.*

NppStatus **nppiNormDiffL2GetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C1R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters.*

NppStatus **nppiNormDiffL2GetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C1R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters.*

NppStatus **nppiNormDiffL2GetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_8u_C1MR_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL2GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_8s_C1MR_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL2GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_16u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_16u_C1MR_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL2GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_32f_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_32f_C1MR_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL2GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_8u_C3R_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL2GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_16u_C3R_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL2GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_16s_C3R_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL2GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_32f_C3R_Ctx`.

For common parameter descriptions, see [*CommonNormDiffL2GetBufferHostSizeParameters*](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_C4R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL2GetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_C4R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL2GetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_C4R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL2GetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_32f_C4R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL2GetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_8u_AC4R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL2GetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16u_AC4R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL2GetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormDiff_L2_16s_AC4R_Ctx.

For common parameter descriptions, see *CommonNormDiffL2GetBufferHostSizeParameters*.

NppStatus **nppiNormDiffL2GetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_32f_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormDiffL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_8u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_8u_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormDiffL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_8s_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_8s_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormDiffL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_16u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_16u_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormDiffL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormDiffL2GetBufferHostSize_32f_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormDiff_L2_32f_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormDiffL2GetBufferHostSizeParameters](#).

1.14.10.8 Image NormRel_Inf

1.14.10.8.1 NormRel_Inf

Primitives for computing the relative error of infinity norm between two images.

Basic NormRel_Inf

NppStatus **nppiNormRel_Inf_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u*
*pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f*
*pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

One-channel 8-bit unsigned image NormRel_Inf.

For common parameter descriptions, see [Common parameters for nppiNorm functions include:](#).

NppStatus **nppiNormRel_Inf_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_16s_C1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_16s_C3R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image NormRel_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image NormRel_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_16s_AC4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image NormRel_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image NormRel_Inf ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_16s_C4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_Inf_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_Inf_8u_C1MR_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit unsigned image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_Inf_8s_C1MR_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit signed image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_Inf_16u_C1MR_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 16-bit unsigned image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_Inf_32f_C1MR_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 32-bit floating point image NormRel_Inf.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_Inf_8u_C3CMR_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit unsigned image NormRel_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_Inf_8s_C3CMR_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit signed image NormRel_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_Inf_16u_C3CMR_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 16-bit unsigned image NormRel_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

```
NppStatus nppiNormRel_Inf_32f_C3CMR_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f
                                         *pSrc2, int nSrc2Step, const Npp8u *pMask, int
                                         nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f
                                         *pNormRel, Npp8u *pDeviceBuffer,
                                         NppStreamContext nppStreamCtx)
```

Masked three-channel 32-bit floating point image NormRel_Inf affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NormRelInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel_Inf primitives.

1.14.10.8.2 CommonNormRelInfGetBufferHostSizeParameters

Common parameters for nppiNormRelInfGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

```
NppStatus nppiNormRelInfGetBufferHostSize_8u_C1R_Ctx(NppiSize oSizeROI, size_t
                                                       *hpBufferSize, NppStreamContext
                                                       nppStreamCtx)
```

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C1R_Ctx.

For common parameter descriptions, see *CommonNormRelInfGetBufferHostSizeParameters.*

```
NppStatus nppiNormRelInfGetBufferHostSize_16u_C1R_Ctx(NppiSize oSizeROI, size_t
                                                         *hpBufferSize,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C1R_Ctx.

For common parameter descriptions, see *CommonNormRelInfGetBufferHostSizeParameters.*

```
NppStatus nppiNormRelInfGetBufferHostSize_16s_C1R_Ctx(NppiSize oSizeROI, size_t
                                                         *hpBufferSize,
                                                         NppStreamContext
                                                         nppStreamCtx)
```


Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16s_C1R_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_32f_C1R_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_8u_C1MR_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_8s_C1MR_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_16u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16u_C1MR_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_32f_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_32f_C1MR_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_8u_C3R_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16u_C3R_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16s_C3R_Ctx.

For common parameter descriptions, see [*CommonNormRelInfGetBufferHostSizeParameters*](#).

NppStatus **nppiNormRelInfGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C3R_Ctx.

For common parameter descriptions, see [*CommonNormRelInfGetBufferHostSizeParameters*](#).

NppStatus **nppiNormRelInfGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_C4R_Ctx.

For common parameter descriptions, see [*CommonNormRelInfGetBufferHostSizeParameters*](#).

NppStatus **nppiNormRelInfGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16u_C4R_Ctx.

For common parameter descriptions, see [*CommonNormRelInfGetBufferHostSizeParameters*](#).

NppStatus **nppiNormRelInfGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_16s_C4R_Ctx.

For common parameter descriptions, see [*CommonNormRelInfGetBufferHostSizeParameters*](#).

NppStatus **nppiNormRelInfGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_32f_C4R_Ctx.

For common parameter descriptions, see [*CommonNormRelInfGetBufferHostSizeParameters*](#).

NppStatus **nppiNormRelInfGetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_Inf_8u_AC4R_Ctx.

For common parameter descriptions, see [*CommonNormRelInfGetBufferHostSizeParameters*](#).

NppStatus **nppiNormRelInfGetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16u_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16s_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_32f_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_8u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_8u_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_8s_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_8s_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_16u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16u_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

NppStatus **nppiNormRelInfGetBufferHostSize_32f_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_32f_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelInfGetBufferHostSizeParameters](#).

1.14.10.9 Image NormRel L1

1.14.10.9.1 NormRel_L1

Primitives for computing the relative error of L1 norm between two images.

Basic NormRel_L1

NppStatus **nppiNormRel_L1_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L1_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L1_16s_C1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L1_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L1_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L1_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormRel_L1_16s_C3R_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s
                                     *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                     aNormRel[3], Npp8u *pDeviceBuffer,
                                     NppStreamContext nppStreamCtx)
```

Three-channel 16-bit signed image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormRel_L1_32f_C3R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f
                                     *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                     aNormRel[3], Npp8u *pDeviceBuffer,
                                     NppStreamContext nppStreamCtx)
```

Three-channel 32-bit floating point image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormRel_L1_8u_AC4R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u
                                     *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                     aNormRel[3], Npp8u *pDeviceBuffer,
                                     NppStreamContext nppStreamCtx)
```

Four-channel 8-bit signed image NormRel_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormRel_L1_16u_AC4R_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u
                                     *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                     aNormRel[3], Npp8u *pDeviceBuffer,
                                     NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned image NormRel_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormRel_L1_16s_AC4R_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s
                                     *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                     aNormRel[3], Npp8u *pDeviceBuffer,
                                     NppStreamContext nppStreamCtx)
```

Four-channel 16-bit signed image NormRel_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormRel_L1_32f_AC4R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f
                                     *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                     aNormRel[3], Npp8u *pDeviceBuffer,
                                     NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating point image NormRel_L1 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormRel_L1_8u_C4R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2,
                                     int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4],
                                     Npp8u *pDeviceBuffer, NppStreamContext
                                     nppStreamCtx)
```

Four-channel 8-bit unsigned image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

```
NppStatus nppiNormRel_L1_16u_C4R_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u
                                     *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                     aNormRel[4], Npp8u *pDeviceBuffer,
                                     NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

```
NppStatus nppiNormRel_L1_16s_C4R_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         aNormRel[4], Npp8u *pDeviceBuffer,
                                         NppStreamContext nppStreamCtx)
```

Four-channel 16-bit signed image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

```
NppStatus nppiNormRel_L1_32f_C4R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         aNormRel[4], Npp8u *pDeviceBuffer,
                                         NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating point image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

```
NppStatus nppiNormRel_L1_8u_C1MR_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u
                                         *pSrc2, int nSrc2Step, const Npp8u *pMask, int
                                         nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel,
                                         Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

One-channel 8-bit unsigned image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

```
NppStatus nppiNormRel_L1_8s_C1MR_Ctx(const Npp8s *pSrc1, int nSrc1Step, const Npp8s
                                         *pSrc2, int nSrc2Step, const Npp8u *pMask, int
                                         nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel,
                                         Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

One-channel 8-bit signed image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

```
NppStatus nppiNormRel_L1_16u_C1MR_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u
                                         *pSrc2, int nSrc2Step, const Npp8u *pMask, int
                                         nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel,
                                         Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

One-channel 16-bit unsigned image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

```
NppStatus nppiNormRel_L1_32f_C1MR_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f
                                         *pSrc2, int nSrc2Step, const Npp8u *pMask, int
                                         nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel,
                                         Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

One-channel 32-bit floating point image NormRel_L1.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNormRel_L1_8u_C3CMR_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit unsigned image NormRel_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_L1_8s_C3CMR_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit signed image NormRel_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_L1_16u_C3CMR_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 16-bit unsigned image NormRel_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_L1_32f_C3CMR_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 32-bit floating point image NormRel_L1 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NormRelL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel_L1 primitives.

1.14.10.9.2 CommonNormRelL1GetBufferHostSizeParameters

Common parameters for nppiNormRelL1GetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes*.

NppStatus **nppiNormRelL1GetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C1R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL1GetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C1R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL1GetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C1R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL1GetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C1R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL1GetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C1MR_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL1GetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8s_C1MR_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL1GetBufferHostSize_16u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C1MR_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL1GetBufferHostSize_32f_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C1MR_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus nppiNormRelL1GetBufferHostSize_8u_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C3R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus nppiNormRelL1GetBufferHostSize_16u_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C3R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus nppiNormRelL1GetBufferHostSize_16s_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C3R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus nppiNormRelL1GetBufferHostSize_32f_C3R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C3R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus nppiNormRelL1GetBufferHostSize_8u_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_8u_C4R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus nppiNormRelL1GetBufferHostSize_16u_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16u_C4R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus nppiNormRelL1GetBufferHostSize_16s_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_16s_C4R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus nppiNormRelL1GetBufferHostSize_32f_C4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L1_32f_C4R_Ctx.

For common parameter descriptions, see *CommonNormRelL1GetBufferHostSizeParameters*.

NppStatus nppiNormRelL1GetBufferHostSize_8u_AC4R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_8u_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelL1GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL1GetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_16u_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelL1GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL1GetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_16s_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelL1GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL1GetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_32f_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelL1GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL1GetBufferHostSize_8u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_8u_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelL1GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL1GetBufferHostSize_8s_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_8s_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelL1GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL1GetBufferHostSize_16u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_16u_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelL1GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL1GetBufferHostSize_32f_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L1_32f_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelL1GetBufferHostSizeParameters](#).

1.14.10.10 Image NormRel_L2

1.14.10.10.1 NormRel_L2

Primitives for computing the relative error of L2 norm between two images.

Basic NormRel_L2

NppStatus **nppiNormRel_L2_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_16s_C1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_16s_C3R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormRel_L2_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormRel_L2_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image NormRel_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormRel_L2_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image NormRel_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormRel_L2_16s_AC4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image NormRel_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormRel_L2_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image NormRel_L2 ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormRel_L2_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus **nppiNormRel_L2_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_16s_C4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aNormRel[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_8u_C1MR_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit unsigned image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_8s_C1MR_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 8-bit signed image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_16u_C1MR_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 16-bit unsigned image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus **nppiNormRel_L2_32f_C1MR_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked one-channel 32-bit floating point image NormRel_L2.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*

NppStatus nppiNormRel_L2_8u_C3CMR_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit unsigned image NormRel_L2 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNormRel_L2_8s_C3CMR_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 8-bit signed image NormRel_L2 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNormRel_L2_16u_C3CMR_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 16-bit unsigned image NormRel_L2 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NppStatus nppiNormRel_L2_32f_C3CMR_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, const *Npp8u* *pMask, int nMaskStep, *NppiSize* oSizeROI, int nCOI, *Npp64f* *pNormRel, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Masked three-channel 32-bit floating point image NormRel_L2 affecting only single channel.

For common parameter descriptions, see *Common parameters for nppiNorm functions include:*.

NormRelL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel_L2 primitives.

1.14.10.10.2 CommonNormRelL2GetBufferHostSizeParameters

Common parameters for nppiNormRelL2GetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes*.

NppStatus **nppiNormRelL2GetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C1R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C1R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C1R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C1R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_8u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C1MR_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_8s_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8s_C1MR_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_16u_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C1MR_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_32f_C1MR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C1MR_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C3R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C3R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C3R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C3R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_8u_C4R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16u_C4R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_16s_C4R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for nppiNormRel_L2_32f_C4R_Ctx.

For common parameter descriptions, see *CommonNormRelL2GetBufferHostSizeParameters*.

NppStatus **nppiNormRelL2GetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_8u_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL2GetBufferHostSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_16u_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL2GetBufferHostSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_16s_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL2GetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_32f_AC4R_Ctx`.

For common parameter descriptions, see [CommonNormRelL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL2GetBufferHostSize_8u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_8u_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL2GetBufferHostSize_8s_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_8s_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL2GetBufferHostSize_16u_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_16u_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelL2GetBufferHostSizeParameters](#).

NppStatus **nppiNormRelL2GetBufferHostSize_32f_C3CMR_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Computes the device scratch buffer size (in bytes) for `nppiNormRel_L2_32f_C3CMR_Ctx`.

For common parameter descriptions, see [CommonNormRelL2GetBufferHostSizeParameters](#).

1.14.11. Image DotProd

1.14.11.1 DotProd

Primitives for computing the dot product of two images.

DotProd

Given two images $pSrc1$ and $pSrc2$ both with width W and height H , the dot product will be computed as

$$DotProd = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [pSrc1(j, i) \cdot pSrc2(j, i)]$$

The functions require additional scratch buffer for computations.

1.14.11.2 Common parameters for nppiDotProd functions include:

param pSrc1

Source-Image Pointer.

param nSrc1Step

Source-Image Line Step.

param pSrc2

Source-Image Pointer.

param nSrc2Step

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pDp

Device memory pointer to the computed dot product of the two images.

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiDotProd_8u64f_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus nppiDotProd_8s64f_C1R_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus nppiDotProd_16u64f_C1R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus nppiDotProd_16s64f_C1R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus nppiDotProd_32u64f_C1R_Ctx(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit unsigned image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus nppiDotProd_32s64f_C1R_Ctx(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit signed image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus nppiDotProd_32f64f_C1R_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus nppiDotProd_8u64f_C3R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus nppiDotProd_8s64f_C3R_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_16u64f_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_16s64f_C3R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_32u64f_C3R_Ctx**(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit unsigned image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_32s64f_C3R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit signed image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_32f64f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_8u64f_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_8s64f_C4R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_16u64f_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*

NppStatus **nppiDotProd_16s64f_C4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*

NppStatus **nppiDotProd_32u64f_C4R_Ctx**(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit unsigned image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*

NppStatus **nppiDotProd_32s64f_C4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*

NppStatus **nppiDotProd_32f64f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[4], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image DotProd.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*

NppStatus **nppiDotProd_8u64f_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image DotProd ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*

NppStatus **nppiDotProd_8s64f_AC4R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image DotProd ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*

NppStatus **nppiDotProd_16u64f_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image DotProd ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*

NppStatus **nppiDotProd_16s64f_AC4R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image DotProd ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_32u64f_AC4R_Ctx**(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit unsigned image DotProd ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_32s64f_AC4R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed image DotProd ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

NppStatus **nppiDotProd_32f64f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* aDp[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image DotProd ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiDotProd functions include:*.

DotProdGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean_StdDev primitives.

1.14.11.3 CommonDotProdGetBufferHostSizeParameters

Common parameters for nppiDotProdGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiDotProdGetBufferHostSize_8u64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C1R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_8s64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C1R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_16u64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C1R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_16s64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C1R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_32u64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C1R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_32s64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C1R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_32f64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C1R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_8u64f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C3R_Ctx.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus nppiDotProdGetBufferHostSize_8s64f_C3R_Ctx([NppiSize](#) oSizeROI, size_t *hpBufferSize, [NppStreamContext](#) nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_8s64f_C3R_Ctx.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus nppiDotProdGetBufferHostSize_16u64f_C3R_Ctx([NppiSize](#) oSizeROI, size_t *hpBufferSize, [NppStreamContext](#) nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_16u64f_C3R_Ctx.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus nppiDotProdGetBufferHostSize_16s64f_C3R_Ctx([NppiSize](#) oSizeROI, size_t *hpBufferSize, [NppStreamContext](#) nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_16s64f_C3R_Ctx.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus nppiDotProdGetBufferHostSize_32u64f_C3R_Ctx([NppiSize](#) oSizeROI, size_t *hpBufferSize, [NppStreamContext](#) nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_32u64f_C3R_Ctx.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus nppiDotProdGetBufferHostSize_32s64f_C3R_Ctx([NppiSize](#) oSizeROI, size_t *hpBufferSize, [NppStreamContext](#) nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_32s64f_C3R_Ctx.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus nppiDotProdGetBufferHostSize_32f64f_C3R_Ctx([NppiSize](#) oSizeROI, size_t *hpBufferSize, [NppStreamContext](#) nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_32f64f_C3R_Ctx.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus nppiDotProdGetBufferHostSize_8u64f_C4R_Ctx([NppiSize](#) oSizeROI, size_t *hpBufferSize, [NppStreamContext](#) nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_8u64f_C4R_Ctx.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus nppiDotProdGetBufferHostSize_8s64f_C4R_Ctx([NppiSize](#) oSizeROI, size_t *hpBufferSize, [NppStreamContext](#) nppStreamCtx)

Device scratch buffer size (in bytes) for `nppiDotProd_8s64f_C4R_Ctx`.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus **nppiDotProdGetBufferHostSize_16u64f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for `nppiDotProd_16u64f_C4R_Ctx`.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus **nppiDotProdGetBufferHostSize_16s64f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for `nppiDotProd_16s64f_C4R_Ctx`.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus **nppiDotProdGetBufferHostSize_32u64f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for `nppiDotProd_32u64f_C4R_Ctx`.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus **nppiDotProdGetBufferHostSize_32s64f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for `nppiDotProd_32s64f_C4R_Ctx`.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus **nppiDotProdGetBufferHostSize_32f64f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for `nppiDotProd_32f64f_C4R_Ctx`.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus **nppiDotProdGetBufferHostSize_8u64f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for `nppiDotProd_8u64f_AC4R_Ctx`.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus **nppiDotProdGetBufferHostSize_8s64f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for `nppiDotProd_8s64f_AC4R_Ctx`.

For common parameter descriptions, see [CommonDotProdGetBufferHostSizeParameters](#).

NppStatus **nppiDotProdGetBufferHostSize_16u64f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_16u64f_AC4R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_16s64f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_16s64f_AC4R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_32u64f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_32u64f_AC4R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_32s64f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_32s64f_AC4R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

NppStatus **nppiDotProdGetBufferHostSize_32f64f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiDotProd_32f64f_AC4R_Ctx.

For common parameter descriptions, see *CommonDotProdGetBufferHostSizeParameters*.

1.14.12. Image CountIn Range

1.14.12.1 CountInRange.

Primitives for computing the amount of pixels that fall into the specified intensity range.

CountInRange

The lower bound and the upper bound are inclusive.

The functions require additional scratch buffer for computations.

1.14.12.2 Common parameters for nppiCountInRange functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pCounts

Device memory pointer to the number of pixels that fall into the specified range.

param nLowerBound

Lower bound of the specified range.

param nUpperBound

Upper bound of the specified range.

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, or NPP_RANGE_ERROR if the lower bound is larger than the upper bound.

NppStatus **nppiCountInRange_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, int *pCounts, *Npp8u* nLowerBound, *Npp8u* nUpperBound, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CountInRange.

For common parameter descriptions, see *Common parameters for nppiCountInRange functions include:*.

NppStatus **nppiCountInRange_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, int *pCounts, *Npp32f* nLowerBound, *Npp32f* nUpperBound, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image CountInRange.

For common parameter descriptions, see *Common parameters for nppiCountInRange functions include:*.

NppStatus **nppiCountInRange_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, int aCounts[3], *Npp8u* aLowerBound[3], *Npp8u* aUpperBound[3], *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CountInRange.

For common parameter descriptions, see *Common parameters for nppiCountInRange functions include:*.

```
NppStatus nppiCountInRange_32f_C3R_Ctx( const Npp32f *pSrc, int nSrcStep, NppiSize
                                         oSizeROI, int aCounts[3], Npp32f aLowerBound[3],
                                         Npp32f aUpperBound[3], Npp8u *pDeviceBuffer,
                                         NppStreamContext nppStreamCtx)
```

Three-channel 32-bit floating point image CountInRange.

For common parameter descriptions, see *Common parameters for nppiCountInRange functions include:*.

```
NppStatus nppiCountInRange_8u_AC4R_Ctx( const Npp8u *pSrc, int nSrcStep, NppiSize oSizeROI,
                                          int aCounts[3], Npp8u aLowerBound[3], Npp8u
                                          aUpperBound[3], Npp8u *pDeviceBuffer,
                                          NppStreamContext nppStreamCtx)
```

Four-channel 8-bit unsigned image CountInRange ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiCountInRange functions include:*.

```
NppStatus nppiCountInRange_32f_AC4R_Ctx( const Npp32f *pSrc, int nSrcStep, NppiSize
                                          oSizeROI, int aCounts[3], Npp32f aLowerBound[3],
                                          Npp32f aUpperBound[3], Npp8u *pDeviceBuffer,
                                          NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating point image CountInRange ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiCountInRange functions include:*.

CountInRangeGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CountInRange primitives.

1.14.12.3 CommonCountInRangeGetBufferHostSizeParameters

Common parameters for nppiCountInRangeGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiCountInRangeGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_C1R_Ctx.

For common parameter descriptions, see *CommonCountInRangeGetBufferHostSizeParameters*.

NppStatus **nppiCountInRangeGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_C1R_Ctx.

For common parameter descriptions, see *CommonCountInRangeGetBufferHostSizeParameters*.

NppStatus **nppiCountInRangeGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_C3R_Ctx.

For common parameter descriptions, see *CommonCountInRangeGetBufferHostSizeParameters*.

NppStatus **nppiCountInRangeGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_C3R_Ctx.

For common parameter descriptions, see *CommonCountInRangeGetBufferHostSizeParameters*.

NppStatus **nppiCountInRangeGetBufferHostSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiCountInRange_8u_AC4R_Ctx.

For common parameter descriptions, see *CommonCountInRangeGetBufferHostSizeParameters*.

NppStatus **nppiCountInRangeGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device scratch buffer size (in bytes) for nppiCountInRange_32f_AC4R_Ctx.

For common parameter descriptions, see *CommonCountInRangeGetBufferHostSizeParameters*.

1.14.13. Image MaxEvery

1.14.13.1 MaxEvery

Primitives for computing the maximal value of the pixel pair from two images.

MaxEvery

The maximum is stored into the second image.

1.14.13.2 Common parameters for nppiMaxEvery functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pSrcDst

In-Place Image Pointer.

param nSrcDstStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMaxEvery_8u_C1IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*

NppStatus **nppiMaxEvery_16u_C1IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*

NppStatus **nppiMaxEvery_16s_C1IR_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_32f_C1IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_8u_C3IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_16u_C3IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_16s_C3IR_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_32f_C3IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_8u_C4IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_16u_C4IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_16s_C4IR_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_32f_C4IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image MaxEvery.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_8u_AC4IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image MaxEvery ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_16u_AC4IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image MaxEvery ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_16s_AC4IR_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image MaxEvery ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

NppStatus **nppiMaxEvery_32f_AC4IR_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image MaxEvery ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMaxEvery functions include:*.

1.14.14. Image MinEvery

1.14.14.1 MinEvery

Primitives for computing the minimal value of the pixel pair from two images.

MinEvery

The minimum is stored into the second image.

1.14.14.2 Common parameters for nppiMinEvery functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param pSrcDst

In-Place Image Pointer.

param nSrcDstStep

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMinEvery_8u_C1IR_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image MinEvery.

For common parameter descriptions, see *Common parameters for nppiMinEvery functions include:*

NppStatus **nppiMinEvery_16u_C1IR_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image MinEvery.

For common parameter descriptions, see *Common parameters for nppiMinEvery functions include:*

NppStatus **nppiMinEvery_16s_C1IR_Ctx**(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image MinEvery.

For common parameter descriptions, see [Common parameters for nppiMinEvery functions include:](#).

NppStatus nppiMinEvery_32f_C1IR_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image MinEvery.

For common parameter descriptions, see [Common parameters for nppiMinEvery functions include:](#).

NppStatus nppiMinEvery_8u_C3IR_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image MinEvery.

For common parameter descriptions, see [Common parameters for nppiMinEvery functions include:](#).

NppStatus nppiMinEvery_16u_C3IR_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image MinEvery.

For common parameter descriptions, see [Common parameters for nppiMinEvery functions include:](#).

NppStatus nppiMinEvery_16s_C3IR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image MinEvery.

For common parameter descriptions, see [Common parameters for nppiMinEvery functions include:](#).

NppStatus nppiMinEvery_32f_C3IR_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image MinEvery.

For common parameter descriptions, see [Common parameters for nppiMinEvery functions include:](#).

NppStatus nppiMinEvery_8u_C4IR_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image MinEvery.

For common parameter descriptions, see [Common parameters for nppiMinEvery functions include:](#).

NppStatus nppiMinEvery_16u_C4IR_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image MinEvery.

For common parameter descriptions, see *Common parameters for nppiMinEvery functions include:*.

NppStatus nppiMinEvery_16s_C4IR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image MinEvery.

For common parameter descriptions, see *Common parameters for nppiMinEvery functions include:*.

NppStatus nppiMinEvery_32f_C4IR_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image MinEvery.

For common parameter descriptions, see *Common parameters for nppiMinEvery functions include:*.

NppStatus nppiMinEvery_8u_AC4IR_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image MinEvery ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinEvery functions include:*.

NppStatus nppiMinEvery_16u_AC4IR_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image MinEvery ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinEvery functions include:*.

NppStatus nppiMinEvery_16s_AC4IR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image MinEvery ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinEvery functions include:*.

NppStatus nppiMinEvery_32f_AC4IR_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image MinEvery ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiMinEvery functions include:*.

1.14.15. Image Integral

1.14.15.1 Integral

Primitives for computing the integral image of a given image.

Integral

Given an input image $pSrc$ and the specified value $nVal$, the pixel value of the integral image $pDst$ at coordinate (i, j) will be computed as

$$pDst(j, i) = nVal + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)$$

If the size of the input image is $W \times H$, the size of the integral image will be $(W + 1) \times (H + 1)$.

NppStatus **nppiIntegral_8u32s_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oROI, *Npp32s* nVal, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image Integral with 32-bit signed output.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oROI** – *Region-Of-Interest (ROI).*
- ▶ **nVal** – The value to add to pDst image pixels
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiIntegral_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oROI, *Npp32f* nVal, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image Integral with 32-bit floating point output.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oROI** – *Region-Of-Interest (ROI).*

- **nVal** – The value to add to pDst image pixels
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

1.14.16. Image Square Integral

1.14.16.1 SqrIntegral

Primitives for computing both the integral and the squared integral images of a given image.

SqrIntegral

Given an input image *pSrc* and the specified value *nVal*, the pixel value of the integral image *pDst* at coordinate (i, j) will be computed as

$$pDst(j, i) = nVal + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)$$

Given an input image *pSrc* and the specified value *nValSqr*, the pixel value of the squared integral image *pSqr* at coordinate (i, j) will be computed as

$$pSqr(j, i) = nValSqr + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)^2$$

If the size of the input image is $W \times H$, the size of the squared integral image will be $(W + 1) \times (H + 1)$.

NppStatus **nppiSqrIntegral_8u32s_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *Npp32s* *pSqr, int nSqrStep, *NppiSize* oSrcROI, *Npp32s* nVal, *Npp32s* nValSqr, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image and square integral image are 32-bit signed int.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*
- **pSqr** – *Destination-Image Pointer.*
- **nSqrStep** – *Destination-Image Line Step.*
- **oSrcROI** – *Region-Of-Interest (ROI).*

- ▶ **nVal** – The value to add to pDst image pixels
- ▶ **nValSqr** – The value to add to pSqr image pixels
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSqrIntegral_8u32s64f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32s* *pDst, int nDstStep, *Npp64f* *pSqr, int nSqrStep, *NppiSize* oSrcROI, *Npp32s* nVal, *Npp64f* nValSqr, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image is 32-bit signed int. Destination square integral image is 64-bit double floating point.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **pSqr** – *Destination-Image Pointer.*
- ▶ **nSqrStep** – *Destination-Image Line Step.*
- ▶ **oSrcROI** – *Region-Of-Interest (ROI).*
- ▶ **nVal** – The value to add to pDst image pixels
- ▶ **nValSqr** – The value to add to pSqr image pixels
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSqrIntegral_8u32f64f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *Npp64f* *pSqr, int nSqrStep, *NppiSize* oSrcROI, *Npp32f* nVal, *Npp64f* nValSqr, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image is 32-bit floating point. Destination square integral image is 64-bit double floating point.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **pSqr** – *Destination-Image Pointer.*

- **nSqrStep** – *Destination-Image Line Step.*
- **oSrcROI** – *Region-Of-Interest (ROI).*
- **nVal** – The value to add to pDst image pixels
- **nValSqr** – The value to add to pSqr image pixels
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.14.17. Image RectStdDev

1.14.17.1 RectStdDev

Primitives for computing the standard deviation of the integral images. The function computes the standard deviation of the pixel in the rectangular window with the integral image *pSrc* and the squared integral image *pSqr*, which can be obtained by calling *Integral* and *SqrIntegral*.

The standard deviation of the pixel (j, i) can be computed using the formula:

$$pDst(j, i) = \sqrt{\max(0, \frac{\sum(SqrIntegral) \cdot N - (\sum(Integral))^2}{N^2})}$$

where $\sum(SqrIntegral) = pSqr[j + oRect.y + oRect.height, i + oRect.x + oRect.width] - pSqr[j + oRect.y, i + oRect.x + oRect.width] - pSqr[j + oRect.y + oRect.height, i + oRect.x] + pSqr[j + oRect.y, i + oRect.x]$, $\sum(Integral) = pSrc[j + oRect.y + oRect.height, i + oRect.x + oRect.width] - pSrc[j + oRect.y, i + oRect.x + oRect.width] - pSrc[j + oRect.y + oRect.height, i + oRect.x] + pSrc[j + oRect.y, i + oRect.x]$, $N = oRect.width \cdot oRect.height$.

The size of the *pSrc* and *pSqr* should be $(oSizeROI.width + oRect.x + oRect.width, oSizeROI.height + oRect.y + oRect.height)$.

RectStdDev

NppStatus **nppiRectStdDev_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp64f* *pSqr, int nSqrStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiRect* oRect, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image RectStdDev.

Parameters

- **pSrc** – *Source-Image Pointer.*
- **nSrcStep** – *Source-Image Line Step.*
- **pSqr** – *Destination-Image Pointer.*
- **nSqrStep** – *Destination-Image Line Step.*
- **pDst** – *Destination-Image Pointer.*
- **nDstStep** – *Destination-Image Line Step.*

- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **oRect** – rectangular window
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRectStdDev_32s_C1RSfs_Ctx**(const *Npp32s* *pSrc, int nSrcStep, const *Npp32s* *pSqr, int nSqrStep, *Npp32s* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiRect* oRect, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One-channel 32-bit signed image RectStdDev, scaled by $2^{(- nScaleFactor)}$.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSqr** – *Destination-Image Pointer.*
- ▶ **nSqrStep** – *Destination-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **oRect** – rectangular window
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiRectStdDev_32s32f_C1R_Ctx**(const *Npp32s* *pSrc, int nSrcStep, const *Npp64f* *pSqr, int nSqrStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppiRect* oRect, *NppStreamContext* nppStreamCtx)

One-channel 32-bit signed image RectStdDev.

Parameters

- ▶ **pSrc** – *Source-Image Pointer.*
- ▶ **nSrcStep** – *Source-Image Line Step.*
- ▶ **pSqr** – *Destination-Image Pointer.*
- ▶ **nSqrStep** – *Destination-Image Line Step.*
- ▶ **pDst** – *Destination-Image Pointer.*
- ▶ **nDstStep** – *Destination-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **oRect** – rectangular window
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Image Data Related Error Codes, ROI Related Error Codes*

1.14.18. Image Histogram Even

1.14.18.1 HistogramEven

Primitives for computing the histogram of an image with evenly distributed bins.

HistogramEven

The *nLowerLevel* (inclusive) and *nUpperLevel* (exclusive) define the boundaries of the range, which are evenly segmented into *nLevel* - 1 bins.

The computed histogram is stored in *pHist*. The levels are calculated by another primitive *nppevenLevelsHost_32s* and are stored in a host pointer *hpLevels*. The number of levels is also *nLevel* - 1. The histogram *pHist[k]* is defined as the total number of pixels that fall into the range: *hpLevels[k]* ≤ *pSrc(j, i)* < *hpLevels[k + 1]*. The functions require additional scratch buffer for computations.

1.14.18.2 Common parameters for nppevenHistogramEven functions include:

param pSrc*Source-Image Pointer.***param nSrcStep***Source-Image Line Step.***param oSizeROI***Region-Of-Interest (ROI).***param pHist**

Device memory pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

param nLevels

Number of levels.

param nLowerLevel

Lower boundary of lowest level bin.

param nUpperLevel

Upper boundary of highest level bin.

param pBuffer

Device memory pointer to appropriately sized scratch buffer.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes.*

NppStatus **nppiEvenLevelsHost_32s**(*Npp32s* *hpLevels, int nLevels, *Npp32s* nLowerLevel, *Npp32s* nUpperLevel)

Compute levels with even distribution.

Parameters

- **hpLevels** – A host pointer to array which receives the levels being computed. The array needs to be of size nLevels.
- **nLevels** – The number of levels being computed. nLevels must be at least 2.
- **nLowerLevel** – Lower boundary value of the lowest level.
- **nUpperLevel** – Upper boundary value of the greatest level.

Returns

image_data_error_codes, or NPP_HISTO_NUMBER_OF_LEVELS_ERROR if an invalid nLevels is specified.

NppStatus **nppiHistogramEven_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist, int nLevels, *Npp32s* nLowerLevel, *Npp32s* nUpperLevel, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned HistogramEven.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus **nppiHistogramEven_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist[3], int nLevels[3], *Npp32s* nLowerLevel[3], *Npp32s* nUpperLevel[3], *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned HistogramEven.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus **nppiHistogramEven_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist[4], int nLevels[4], *Npp32s* nLowerLevel[4], *Npp32s* nUpperLevel[4], *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned HistogramEven.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus **nppiHistogramEven_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist[3], int nLevels[3], *Npp32s* nLowerLevel[3], *Npp32s* nUpperLevel[3], *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned HistogramEven ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus **nppiHistogramEven_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist, int nLevels, *Npp32s* nLowerLevel, *Npp32s* nUpperLevel, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned HistogramEven.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramEven_16u_C3R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiSize
                                         oSizeROI, Npp32s *pHist[3], int nLevels[3], Npp32s
                                         nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Three-channel 16-bit unsigned HistogramEven.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramEven_16u_C4R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiSize
                                         oSizeROI, Npp32s *pHist[4], int nLevels[4], Npp32s
                                         nLowerLevel[4], Npp32s nUpperLevel[4], Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned HistogramEven.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramEven_16u_AC4R_Ctx( const Npp16u *pSrc, int nSrcStep, NppiSize
                                         oSizeROI, Npp32s *pHist[3], int nLevels[3],
                                         Npp32s nLowerLevel[3], Npp32s nUpperLevel[3],
                                         Npp8u *pBuffer, NppStreamContext
                                         nppStreamCtx)
```

Four-channel 16-bit unsigned HistogramEven ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramEven_16s_C1R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize
                                         oSizeROI, Npp32s *pHist, int nLevels, Npp32s
                                         nLowerLevel, Npp32s nUpperLevel, Npp8u *pBuffer,
                                         NppStreamContext nppStreamCtx)
```

One-channel 16-bit signed HistogramEven.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramEven_16s_C3R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize
                                         oSizeROI, Npp32s *pHist[3], int nLevels[3], Npp32s
                                         nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Three-channel 16-bit signed HistogramEven.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramEven_16s_C4R_Ctx( const Npp16s *pSrc, int nSrcStep, NppiSize
                                         oSizeROI, Npp32s *pHist[4], int nLevels[4], Npp32s
                                         nLowerLevel[4], Npp32s nUpperLevel[4], Npp8u
                                         *pBuffer, NppStreamContext nppStreamCtx)
```

Four-channel 16-bit signed HistogramEven.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramEven_16s_AC4R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize
                                         oSizeROI, Npp32s *pHist[3], int nLevels[3],
                                         Npp32s nLowerLevel[3], Npp32s nUpperLevel[3],
                                         Npp8u *pBuffer, NppStreamContext
                                         nppStreamCtx)
```

Four-channel 16-bit signed HistogramEven ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

HistogramEvenGetBufferSize

Companion primitives for computing the device buffer size (in bytes) required by the HistogramEven primitives.

1.14.18.3 Common parameters for nppiHistogramEvenGetBufferSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param nLevels

Number of levels in the histogram.

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes..*

```
NppStatus nppiHistogramEvenGetBufferSize_8u_C1R_Ctx(NppiSize oSizeROI, int nLevels,
                                                    size_t *hpBufferSize,
                                                    NppStreamContext nppStreamCtx)
```

Buffer size for *nppiHistogramEven_8u_C1R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

```
NppStatus nppiHistogramEvenGetBufferSize_8u_C3R_Ctx(NppiSize oSizeROI, int nLevels[3],
                                                    size_t *hpBufferSize,
                                                    NppStreamContext nppStreamCtx)
```

Buffer size for *nppiHistogramEven_8u_C3R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

```
NppStatus nppiHistogramEvenGetBufferSize_8u_C4R_Ctx(NppiSize oSizeROI, int nLevels[4],
                                                    size_t *hpBufferSize,
                                                    NppStreamContext nppStreamCtx)
```

Buffer size for *nppiHistogramEven_8u_C4R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramEvenGetBufferSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, int nLevels[3],
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiHistogramEven_8u_AC4R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramEvenGetBufferSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, int nLevels,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiHistogramEven_16u_C1R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramEvenGetBufferSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, int nLevels[3],
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiHistogramEven_16u_C3R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramEvenGetBufferSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, int nLevels[4],
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiHistogramEven_16u_C4R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramEvenGetBufferSize_16u_AC4R_Ctx**(*NppiSize* oSizeROI, int nLevels[3],
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiHistogramEven_16u_AC4R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramEvenGetBufferSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, int nLevels,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiHistogramEven_16s_C1R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramEvenGetBufferSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, int nLevels[3], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiHistogramEven_16s_C3R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramEvenGetBufferSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, int nLevels[4], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiHistogramEven_16s_C4R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramEvenGetBufferSize_16s_AC4R_Ctx**(*NppiSize* oSizeROI, int nLevels[3], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiHistogramEven_16s_AC4R_Ctx*.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

1.14.19. Image Histogram Range

1.14.19.1 HistogramRange

Primitives for computing the histogram of an image within specified ranges.

HistogramRange

The histogram is computed according to the ranges provided in *pLevels*.

The histogram $pHist[k]$ is defined as the total number of pixels that fall into the range: $pLevels[k] \leq pSrc(j, i) < pLevels[k + 1]$. The number of the histogram bins is $nLevel - 1$. The functions require additional scratch buffer for computations.

1.14.19.2 Common parameters for nppiHistogramRange functions include:

- param pSrc**
Source-Image Pointer.
- param nSrcStep**
Source-Image Line Step.
- param oSizeROI**
Region-Of-Interest (ROI).

param pHist

Device memory pointer to array that receives the computed histogram. The array must be of size nLevels-1.

param pLevels

Device memory pointer to array containing the level sizes of the bins. The array must be of size nLevels.

param nLevels

Number of levels in histogram.

param pBuffer

Pointer to appropriately sized (nppiHistogramRangeGetBufferSize_XX_XXX) scratch buffer.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiHistogramRange_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist, const *Npp32s* *pLevels, int nLevels, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus **nppiHistogramRange_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist[3], const *Npp32s* *pLevels[3], int nLevels[3], *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus **nppiHistogramRange_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist[4], const *Npp32s* *pLevels[4], int nLevels[4], *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus **nppiHistogramRange_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist[3], const *Npp32s* *pLevels[3], int nLevels[3], *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned HistogramRange ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus **nppiHistogramRange_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist, const *Npp32s* *pLevels, int nLevels, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramRange_16u_C3R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize
oSizeROI, Npp32s *pHist[3], const Npp32s
*pLevels[3], int nLevels[3], Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Three-channel 16-bit unsigned HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramRange_16u_C4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize
oSizeROI, Npp32s *pHist[4], const Npp32s
*pLevels[4], int nLevels[4], Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramRange_16u_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize
oSizeROI, Npp32s *pHist[3], const Npp32s
*pLevels[3], int nLevels[3], Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Four-channel 16-bit unsigned HistogramRange ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramRange_16s_C1R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize
oSizeROI, Npp32s *pHist, const Npp32s *pLevels,
int nLevels, Npp8u *pBuffer, NppStreamContext
nppStreamCtx)
```

One-channel 16-bit signed HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramRange_16s_C3R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize
oSizeROI, Npp32s *pHist[3], const Npp32s
*pLevels[3], int nLevels[3], Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Three-channel 16-bit signed HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

```
NppStatus nppiHistogramRange_16s_C4R_Ctx(const Npp16s *pSrc, int nSrcStep, NppiSize
oSizeROI, Npp32s *pHist[4], const Npp32s
*pLevels[4], int nLevels[4], Npp8u *pBuffer,
NppStreamContext nppStreamCtx)
```

Four-channel 16-bit signed HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus nppiHistogramRange_16s_AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist[3], const *Npp32s* *pLevels[3], int nLevels[3], *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus nppiHistogramRange_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist, const *Npp32f* *pLevels, int nLevels, *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus nppiHistogramRange_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist[3], const *Npp32f* *pLevels[3], int nLevels[3], *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus nppiHistogramRange_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist[4], const *Npp32f* *pLevels[4], int nLevels[4], *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point HistogramRange.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

NppStatus nppiHistogramRange_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, *Npp32s* *pHist[3], const *Npp32f* *pLevels[3], int nLevels[3], *Npp8u* *pBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point HistogramRange ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiHistogramEven functions include:*.

HistogramRangeGetBufferSize

Companion primitives for computing the device buffer size (in bytes) required by the HistogramRange primitives.

NppStatus nppiHistogramRangeGetBufferSize_8u_C1R_Ctx(*NppiSize* oSizeROI, int nLevels, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_8u_C1R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramRangeGetBufferSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, int nLevels[3], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_8u_C3R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramRangeGetBufferSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, int nLevels[4], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_8u_C4R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramRangeGetBufferSize_8u_AC4R_Ctx**(*NppiSize* oSizeROI, int nLevels[3], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_8u_AC4R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramRangeGetBufferSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, int nLevels, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_16u_C1R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramRangeGetBufferSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, int nLevels[3], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_16u_C3R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus **nppiHistogramRangeGetBufferSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, int nLevels[4], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_16u_C4R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus nppiHistogramRangeGetBufferSize_16u_AC4R_Ctx(*NppiSize* oSizeROI, int nLevels[3], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_16u_AC4R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus nppiHistogramRangeGetBufferSize_16s_C1R_Ctx(*NppiSize* oSizeROI, int nLevels, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_16s_C1R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus nppiHistogramRangeGetBufferSize_16s_C3R_Ctx(*NppiSize* oSizeROI, int nLevels[3], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_16s_C3R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus nppiHistogramRangeGetBufferSize_16s_C4R_Ctx(*NppiSize* oSizeROI, int nLevels[4], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_16s_C4R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus nppiHistogramRangeGetBufferSize_16s_AC4R_Ctx(*NppiSize* oSizeROI, int nLevels[3], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_16s_AC4R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus nppiHistogramRangeGetBufferSize_32f_C1R_Ctx(*NppiSize* oSizeROI, int nLevels, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for nppiHistogramRange_32f_C1R_Ctx.

For common parameter descriptions, see *Common parameters for nppiHistogramEvenGetBufferSize functions include:*.

NppStatus nppiHistogramRangeGetBufferSize_32f_C3R_Ctx(*NppiSize* oSizeROI, int nLevels[3], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for `nppiHistogramRange_32f_C3R_Ctx`.

For common parameter descriptions, see [Common parameters for nppiHistogramEvenGetBufferSize functions include:](#).

NppStatus `nppiHistogramRangeGetBufferSize_32f_C4R_Ctx`(*NppSize* oSizeROI, int nLevels[4], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for `nppiHistogramRange_32f_C4R_Ctx`.

For common parameter descriptions, see [Common parameters for nppiHistogramEvenGetBufferSize functions include:](#).

NppStatus `nppiHistogramRangeGetBufferSize_32f_AC4R_Ctx`(*NppSize* oSizeROI, int nLevels[3], size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Scratch-buffer size for `nppiHistogramRange_32f_AC4R_Ctx`.

For common parameter descriptions, see [Common parameters for nppiHistogramEvenGetBufferSize functions include:](#).

1.14.20. Image Proximity

1.14.20.1 Image Proximity

Primitives for computing the proximity measure between a source image and a template image.

1.14.20.1.1 General Introduction

There are basically two approaches to compute the proximity measure for template matching, Euclidean distance and the cross correlation.

1.

Euclidean distance computes the sum of the squared distance (SSD) between the corresponding pixels of the source image and the template image. The smaller the distance is, the more similar the source image and the template image is around the pixel. The anchor of the template

image is used during the computations, which always lies in the geometric center of the image. Given a source image $pSrc$ ($W_s \times H_s$) and a template image $pTpl$ ($W_t \times H_t$), the Euclidean distance $D_{st}(c, r)$ between two images at pixel in row r and column c is computed as (s stands for source image and t for template image for short):

$$D_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2})]^2$$

2. Cross correlation computes the sum of the product between the corresponding pixels of the source image and the template image. The cross correlation $R_{st}(c, r)$ is calculated as:

$$R_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) \cdot pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2})]$$

The larger the cross correlation value is, the more similar the source image and the template image is around the pixel.

3. The cross correlation $R_{st}(c, r)$ is affected by the brightness of the images which may vary due to the lighting and exposure conditions. Therefore, NPP computes the cross correlation coefficient to circumvent this dependence. This is typically done at every step by subtracting the mean from every pixel value, i.e.,

$$\tilde{R}_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - Mean_t] \cdot [pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2}) - Mean_s]$$

NPP computes the normalized values of Euclidean distance, cross correlation and the cross correlation coefficient.

1. The normalized Euclidean distance $\sigma_{st}(c, r)$ is defined as:

$$\sigma_{st}(c, r) = \frac{D_{st}(c, r)}{\sqrt{R_{ss}(c, r) \cdot R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

2. The normalized cross correlation $\rho_{st}(c, r)$ is defined as:

$$\rho_{st}(c, r) = \frac{R_{st}(c, r)}{\sqrt{R_{ss}(c, r) \cdot R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

The $R_{ss}(c, r)$ and $R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})$ denote the auto correlation of the source image and the template image individually. They are defined as:

$$R_{ss}(c, r) = \sum_{j=c-\frac{H_t}{2}}^{c+\frac{H_t}{2}} \sum_{i=r-\frac{W_t}{2}}^{r+\frac{W_t}{2}} pSrc(j, i)$$

$$R_{tt}(\frac{H_t}{2}, \frac{W_t}{2}) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} pTpl(j, i)$$

3. Similarly, the normalized cross correlation coefficient $\gamma_{st}(c, r)$ is calculated as:

$$\gamma_{st}(c, r) = \frac{\tilde{R}_{st}(c, r)}{\sqrt{\tilde{R}_{ss}(c, r) \cdot \tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

The $\tilde{R}_{ss}(c, r)$ and $\tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2})$ are defined as:

$$\tilde{R}_{ss}(c, r) = \sum_{j=c-\frac{H_t}{2}}^{c+\frac{H_t}{2}} \sum_{i=r-\frac{W_t}{2}}^{r+\frac{W_t}{2}} [pSrc(j, i) - Mean_s]$$

$$\tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2}) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - Mean_t]$$

where $Mean_t$ is the template mean minus the mean of the image in the region just under the template.

1.14.20.1.2 Categorizations

The Euclidean distance and the cross correlation are categorized into three types, full, same, and valid.

1. Full mode indicates that the anchor of the template image starts from the outside of the source image, assuming the out-of-boundary pixels are zero-padded. The size of the destination image is $(W_s + W_t - 1) \times (H_s + H_t - 1)$.
2. Same mode means that the anchor of the template image starts from the top left pixel of the source image. All the out-of-boundary pixels are also zero-padded. The size of the destination image is the same as the source one, i.e., $W_s \times H_s$.
3. Valid mode indicates that there are no out-of-boundary readings from the source image. The anchor of the template image starts from the inside of the source image. The size of the destination image is $(W_s - W_t + 1) \times (H_s - H_t + 1)$.

1.14.21. Image Square Distance Full Norm

1.14.21.1 SqrDistanceFull_Norm

Primitives for computing the normalized Euclidean distance between two images with full mode.

SqrDistanceFull_Norm

The functions compute the $\sigma_{st}(c, r)$ in *General Introduction* with full mode (see *Categorizations*).

1.14.21.2 Common parameters for nppiSqrDistanceFull functions include:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param oSrcRoiSize

Region-Of-Interest (ROI).

param pTpl

Pointer to the template image.

param nTplStep

Number of bytes between successive rows in the template image.

param oTplRoiSize

Region-Of-Interest (ROI).

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param nScaleFactor

Integer Result Scaling.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

NppStatus nppiSqrDistanceFull_Norm_8u_C1RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{\text{--}nScaleFactor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceFull_Norm_8u_C3RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{\text{--}nScaleFactor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceFull_Norm_8u_C4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceFull_Norm, scaled by $2^{\text{--}nScaleFactor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceFull_Norm_8u_AC4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel, scaled by $2^{\text{--}nScaleFactor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceFull_Norm_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiSqrDistanceFull_Norm_32f_C3R_Ctx( const Npp32f *pSrc, int nSrcStep,  
                                                NppiSize oSrcRoiSize, const Npp32f *pTpl,  
                                                int nTplStep, NppiSize oTplRoiSize, Npp32f  
                                                *pDst, int nDstStep, NppStreamContext  
                                                nppStreamCtx )
```

Three-channel 32-bit floating point image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiSqrDistanceFull_Norm_32f_C4R_Ctx( const Npp32f *pSrc, int nSrcStep,  
                                                NppiSize oSrcRoiSize, const Npp32f *pTpl,  
                                                int nTplStep, NppiSize oTplRoiSize, Npp32f  
                                                *pDst, int nDstStep, NppStreamContext  
                                                nppStreamCtx )
```

Four-channel 32-bit floating point image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiSqrDistanceFull_Norm_32f_AC4R_Ctx( const Npp32f *pSrc, int nSrcStep,  
                                                NppiSize oSrcRoiSize, const Npp32f  
                                                *pTpl, int nTplStep, NppiSize oTplRoiSize,  
                                                Npp32f *pDst, int nDstStep,  
                                                NppStreamContext nppStreamCtx )
```

Four-channel 32-bit floating point image SqrDistanceFull_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiSqrDistanceFull_Norm_8u32f_C1R_Ctx( const Npp8u *pSrc, int nSrcStep,  
                                                NppiSize oSrcRoiSize, const Npp8u  
                                                *pTpl, int nTplStep, NppiSize  
                                                oTplRoiSize, Npp32f *pDst, int  
                                                nDstStep, NppStreamContext  
                                                nppStreamCtx )
```

One-channel 8-bit unsigned image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiSqrDistanceFull_Norm_8u32f_C3R_Ctx( const Npp8u *pSrc, int nSrcStep,  
                                                NppiSize oSrcRoiSize, const Npp8u  
                                                *pTpl, int nTplStep, NppiSize  
                                                oTplRoiSize, Npp32f *pDst, int  
                                                nDstStep, NppStreamContext  
                                                nppStreamCtx )
```

Three-channel 8-bit unsigned image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceFull_Norm_8u32f_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceFull_Norm_8u32f_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceFull_Norm_8s32f_C1R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceFull_Norm_8s32f_C3R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceFull_Norm_8s32f_C4R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceFull_Norm_8s32f_AC4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image SqrDistanceFull_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceFull_Norm_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceFull_Norm_16u32f_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceFull_Norm_16u32f_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image SqrDistanceFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceFull_Norm_16u32f_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image SqrDistanceFull_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

1.14.22. Image Square Distance Same Norm

1.14.22.1 SqrDistanceSame_Norm

Primitives for computing the normalized Euclidean distance between two images with same mode.

SqrDistanceSame_Norm

The functions compute the $\sigma_{st}(c, r)$ in *General Introduction* with same mode (see *Categorizations*).

NppStatus nppiSqrDistanceSame_Norm_8u_C1RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\ell - nScaleFactor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceSame_Norm_8u_C3RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\ell - nScaleFactor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceSame_Norm_8u_C4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceSame_Norm, scaled by $2^{\ell - nScaleFactor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceSame_Norm_8u_AC4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel, scaled by $2^{\ell - nScaleFactor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image SqrDistanceSame_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_8u32f_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_8u32f_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_8u32f_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_8s32f_C1R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_8s32f_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_8s32f_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiSqrDistanceSame_Norm_8s32f_AC4R_Ctx(const Npp8s *pSrc, int nSrcStep,  
                                                    NppiSize oSrcRoiSize, const Npp8s  
                                                    *pTpl, int nTplStep, NppiSize  
                                                    oTplRoiSize, Npp32f *pDst, int  
                                                    nDstStep, NppStreamContext  
                                                    nppStreamCtx)
```

Four-channel 8-bit signed image SqrDistanceSame_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiSqrDistanceSame_Norm_16u32f_C1R_Ctx(const Npp16u *pSrc, int nSrcStep,  
                                                    NppiSize oSrcRoiSize, const Npp16u  
                                                    *pTpl, int nTplStep, NppiSize  
                                                    oTplRoiSize, Npp32f *pDst, int  
                                                    nDstStep, NppStreamContext  
                                                    nppStreamCtx)
```

One-channel 16-bit unsigned image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiSqrDistanceSame_Norm_16u32f_C3R_Ctx(const Npp16u *pSrc, int nSrcStep,  
                                                    NppiSize oSrcRoiSize, const Npp16u  
                                                    *pTpl, int nTplStep, NppiSize  
                                                    oTplRoiSize, Npp32f *pDst, int  
                                                    nDstStep, NppStreamContext  
                                                    nppStreamCtx)
```

Three-channel 16-bit unsigned image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiSqrDistanceSame_Norm_16u32f_C4R_Ctx(const Npp16u *pSrc, int nSrcStep,  
                                                    NppiSize oSrcRoiSize, const Npp16u  
                                                    *pTpl, int nTplStep, NppiSize  
                                                    oTplRoiSize, Npp32f *pDst, int  
                                                    nDstStep, NppStreamContext  
                                                    nppStreamCtx)
```

Four-channel 16-bit unsigned image SqrDistanceSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiSqrDistanceSame_Norm_16u32f_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep,  
                                                    NppiSize oSrcRoiSize, const Npp16u  
                                                    *pTpl, int nTplStep, NppiSize  
                                                    oTplRoiSize, Npp32f *pDst, int  
                                                    nDstStep, NppStreamContext  
                                                    nppStreamCtx)
```

Four-channel 16-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceSame_Norm_16u32f_AC4R**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceSame_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

1.14.23. Image Square Distance Valid Norm

1.14.23.1 SqrDistanceValid_Norm

Primitives for computing the normalized Euclidean distance between two images with valid mode.

SqrDistanceValid_Norm

The functions compute the $\sigma_{st}(c, r)$ in *General Introduction* with valid mode (see *Categorizations*).

NppStatus **nppiSqrDistanceValid_Norm_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{(- nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceValid_Norm_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{(- nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceValid_Norm_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceValid_Norm, scaled by $2^{(- nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_8u_AC4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel, scaled by $2^{(nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image SqrDistanceValid_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_8u32f_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_8u32f_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_8u32f_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_8u32f_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_8s32f_C1R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceValid_Norm_8s32f_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceValid_Norm_8s32f_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceValid_Norm_8s32f_AC4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image SqrDistanceValid_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceValid_Norm_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiSqrDistanceValid_Norm_16u32f_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_16u32f_C4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image SqrDistanceValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiSqrDistanceValid_Norm_16u32f_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image SqrDistanceValid_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

1.14.24. Image Cross Correlation Full Norm

1.14.24.1 CrossCorrFull_Norm

Primitives for computing the normalized cross correlation between two images with full mode.

CrossCorrFull_Norm

The functions compute the $\rho_{st}(c, r)$ in *General Introduction* with full mode (see *Categorizations*).

NppStatus nppiCrossCorrFull_Norm_8u_C1RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{(-nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_Norm_8u_C3RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{(-nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_Norm_8u_C4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrFull_Norm, scaled by $2^{(- nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_Norm_8u_AC4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrFull_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_Norm_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_Norm_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_Norm_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_Norm_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrFull_Norm ignoring alpha channel.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus **nppiCrossCorrFull_Norm_64f_C1R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point image CrossCorrFull_Norm.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus **nppiCrossCorrFull_Norm_64f_C3R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 64-bit floating point image CrossCorrFull_Norm.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus **nppiCrossCorrFull_Norm_64f_C4R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrFull_Norm.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus **nppiCrossCorrFull_Norm_64f_AC4R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrFull_Norm ignoring alpha channel.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus **nppiCrossCorrFull_Norm_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrFull_Norm.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus **nppiCrossCorrFull_Norm_8u32f_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_Norm_8u32f_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_Norm_8u32f_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrFull_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_Norm_8s32f_C1R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_Norm_8s32f_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_Norm_8s32f_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrFull_Norm_8s32f_AC4R_Ctx( const Npp8s *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp8s *pTpl,
                                                    int nTplStep, NppiSize oTplRoiSize,
                                                    Npp32f *pDst, int nDstStep,
                                                    NppStreamContext nppStreamCtx )
```

Four-channel 8-bit signed image CrossCorrFull_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrFull_Norm_16u32f_C1R_Ctx( const Npp16u *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp16u
                                                    *pTpl, int nTplStep, NppiSize oTplRoiSize,
                                                    Npp32f *pDst, int nDstStep,
                                                    NppStreamContext nppStreamCtx )
```

One-channel 16-bit unsigned image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrFull_Norm_16u32f_C3R_Ctx( const Npp16u *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp16u
                                                    *pTpl, int nTplStep, NppiSize oTplRoiSize,
                                                    Npp32f *pDst, int nDstStep,
                                                    NppStreamContext nppStreamCtx )
```

Three-channel 16-bit unsigned image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrFull_Norm_16u32f_C4R_Ctx( const Npp16u *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp16u
                                                    *pTpl, int nTplStep, NppiSize oTplRoiSize,
                                                    Npp32f *pDst, int nDstStep,
                                                    NppStreamContext nppStreamCtx )
```

Four-channel 16-bit unsigned image CrossCorrFull_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrFull_Norm_16u32f_AC4R_Ctx( const Npp16u *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp16u
                                                    *pTpl, int nTplStep, NppiSize
                                                    oTplRoiSize, Npp32f *pDst, int
                                                    nDstStep, NppStreamContext
                                                    nppStreamCtx )
```

Four-channel 16-bit unsigned image CrossCorrFull_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

1.14.25. Image Cross Correlation Same Norm

1.14.25.1 CrossCorrSame_Norm

Primitives for computing the normalized cross correlation between two images with same mode.

CrossCorrSame_Norm

The functions compute the $\rho_{st}(c, r)$ in *General Introduction* with same mode (see *Categorizations*).

NppStatus nppiCrossCorrSame_Norm_8u_C1RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\lfloor -nScaleFactor \rfloor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_8u_C3RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\lfloor -nScaleFactor \rfloor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_8u_C4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrSame_Norm, scaled by $2^{\lfloor -nScaleFactor \rfloor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_8u_AC4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrSame_Norm ignoring alpha channel, scaled by $2^{\lfloor -nScaleFactor \rfloor}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrSame_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_64f_C1R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_64f_C3R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 64-bit floating point image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_64f_C4R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_64f_AC4R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrSame_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_8u32f_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_8u32f_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_8u32f_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrSame_Norm_8u32f_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrSame_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_Norm_8s32f_C1R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_Norm_8s32f_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_Norm_8s32f_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_Norm_8s32f_AC4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image CrossCorrSame_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_Norm_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_Norm_16u32f_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_Norm_16u32f_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image CrossCorrSame_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_Norm_16u32f_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image CrossCorrSame_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

1.14.26. Image Cross Correlation Valid Norm

1.14.26.1 CrossCorrValid_Norm

Primitives for computing the normalized cross correlation between two images with valid mode.

CrossCorrValid_Norm

The functions compute the $\rho_{st}(c, r)$ in *General Introduction* with valid mode (see *Categorizations*).

NppStatus **nppiCrossCorrValid_Norm_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{(-nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_8u_C3RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{(- nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_8u_C4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrValid_Norm, scaled by $2^{(- nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_8u_AC4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrValid_Norm ignoring alpha channel, scaled by $2^{(- nScaleFactor)}$.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrValid_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_64f_C1R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_64f_C3R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Three-channel 64-bit floating point image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_64f_C4R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_64f_AC4R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrValid_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_8u32f_C1R_Ctx( const Npp8u *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp8u *pTpl,
                                                    int nTplStep, NppiSize oTplRoiSize,
                                                    Npp32f *pDst, int nDstStep,
                                                    NppStreamContext nppStreamCtx )
```

One-channel 8-bit unsigned image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_8u32f_C3R_Ctx( const Npp8u *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp8u *pTpl,
                                                    int nTplStep, NppiSize oTplRoiSize,
                                                    Npp32f *pDst, int nDstStep,
                                                    NppStreamContext nppStreamCtx )
```

Three-channel 8-bit unsigned image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_8u32f_C4R_Ctx( const Npp8u *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp8u *pTpl,
                                                    int nTplStep, NppiSize oTplRoiSize,
                                                    Npp32f *pDst, int nDstStep,
                                                    NppStreamContext nppStreamCtx )
```

Four-channel 8-bit unsigned image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_8u32f_AC4R_Ctx( const Npp8u *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp8u
                                                    *pTpl, int nTplStep, NppiSize
                                                    oTplRoiSize, Npp32f *pDst, int
                                                    nDstStep, NppStreamContext
                                                    nppStreamCtx )
```

Four-channel 8-bit unsigned image CrossCorrValid_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_8s32f_C1R_Ctx( const Npp8s *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp8s *pTpl,
                                                    int nTplStep, NppiSize oTplRoiSize,
                                                    Npp32f *pDst, int nDstStep,
                                                    NppStreamContext nppStreamCtx )
```

One-channel 8-bit signed image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_8s32f_C3R_Ctx( const Npp8s *pSrc, int nSrcStep,
                                                    NppiSize oSrcRoiSize, const Npp8s *pTpl,
                                                    int nTplStep, NppiSize oTplRoiSize,
                                                    Npp32f *pDst, int nDstStep,
                                                    NppStreamContext nppStreamCtx )
```


Three-channel 8-bit signed image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_8s32f_C4R_Ctx( const Npp8s *pSrc, int nSrcStep,  
                                                  NppiSize oSrcRoiSize, const Npp8s *pTpl,  
                                                  int nTplStep, NppiSize oTplRoiSize,  
                                                  Npp32f *pDst, int nDstStep,  
                                                  NppStreamContext nppStreamCtx)
```

Four-channel 8-bit signed image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_8s32f_AC4R_Ctx( const Npp8s *pSrc, int nSrcStep,  
                                                  NppiSize oSrcRoiSize, const Npp8s  
                                                  *pTpl, int nTplStep, NppiSize  
                                                  oTplRoiSize, Npp32f *pDst, int  
                                                  nDstStep, NppStreamContext  
                                                  nppStreamCtx)
```

Four-channel 8-bit signed image CrossCorrValid_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_16u32f_C1R_Ctx( const Npp16u *pSrc, int nSrcStep,  
                                                  NppiSize oSrcRoiSize, const Npp16u  
                                                  *pTpl, int nTplStep, NppiSize  
                                                  oTplRoiSize, Npp32f *pDst, int  
                                                  nDstStep, NppStreamContext  
                                                  nppStreamCtx)
```

One-channel 16-bit unsigned image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_16u32f_C3R_Ctx( const Npp16u *pSrc, int nSrcStep,  
                                                  NppiSize oSrcRoiSize, const Npp16u  
                                                  *pTpl, int nTplStep, NppiSize  
                                                  oTplRoiSize, Npp32f *pDst, int  
                                                  nDstStep, NppStreamContext  
                                                  nppStreamCtx)
```

Three-channel 16-bit unsigned image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_Norm_16u32f_C4R_Ctx( const Npp16u *pSrc, int nSrcStep,  
                                                  NppiSize oSrcRoiSize, const Npp16u  
                                                  *pTpl, int nTplStep, NppiSize  
                                                  oTplRoiSize, Npp32f *pDst, int  
                                                  nDstStep, NppStreamContext  
                                                  nppStreamCtx)
```

Four-channel 16-bit unsigned image CrossCorrValid_Norm.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_Norm_16u32f_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image CrossCorrValid_Norm ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

1.14.27. Image Cross Correlation Valid

1.14.27.1 CrossCorrValid

Primitives for computing the cross correlation between two images with valid mode.

CrossCorrValid

The functions compute the $R_{st}(c, r)$ in *General Introduction* with valid mode (see *Categorizations*).

NppStatus nppiCrossCorrValid_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point images CrossCorrValid.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_64f_C1R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point images CrossCorrValid.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_8u32f_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned images CrossCorrValid.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_8s32f_C1R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed images CrossCorrValid.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

```
NppStatus nppiCrossCorrValid_16u32f_C1R_Ctx(const Npp16u *pSrc, int nSrcStep, NppiSize
                                             oSrcRoiSize, const Npp16u *pTpl, int nTplStep,
                                             NppiSize oTplRoiSize, Npp32f *pDst, int
                                             nDstStep, NppStreamContext nppStreamCtx)
```

One-channel 16-bit unsigned images CrossCorrValid.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

1.14.28. Image Cross Correlation Full Norm Level

1.14.28.1 CrossCorrFull_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with full mode.

CrossCorrFull_NormLevel

The functions compute the $\gamma_{st}(c, r)$ in [General Introduction](#) with full mode (see [Categorizations](#)).

The functions require additional scratch buffer for computations.

Note: For maximum performance oSrcRoiSize.width + oTplRoiSize.width - 1 MUST be an integer multiple of 4.

```
NppStatus nppiCrossCorrFull_NormLevel_8u_C1RSfs_Ctx(const Npp8u *pSrc, int nSrcStep,
                                                       NppiSize oSrcRoiSize, const Npp8u
                                                       *pTpl, int nTplStep, NppiSize
                                                       oTplRoiSize, Npp8u *pDst, int
                                                       nDstStep, int nScaleFactor, Npp8u
                                                       *pDeviceBuffer, NppStreamContext
                                                       nppStreamCtx)
```

One-channel 8-bit unsigned image CrossCorrFull_NormLevel.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

```
NppStatus nppiCrossCorrFull_NormLevel_8u_C3RSfs_Ctx(const Npp8u *pSrc, int nSrcStep,
                                                       NppiSize oSrcRoiSize, const Npp8u
                                                       *pTpl, int nTplStep, NppiSize
                                                       oTplRoiSize, Npp8u *pDst, int
                                                       nDstStep, int nScaleFactor, Npp8u
                                                       *pDeviceBuffer, NppStreamContext
                                                       nppStreamCtx)
```

Three-channel 8-bit unsigned image CrossCorrFull_NormLevel.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus nppiCrossCorrFull_NormLevel_8u_C4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_NormLevel_8u_AC4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_NormLevel_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_NormLevel_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_NormLevel_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrFull_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_64f_C1R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_64f_C3R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 64-bit floating point image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_64f_C4R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_64f_AC4R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrFull_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_NormLevel1_8u32f_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_NormLevel1_8u32f_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_NormLevel1_8u32f_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_NormLevel1_8u32f_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrFull_NormLevel1_8s32f_C1R_Ctx(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_8s32f_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_8s32f_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_8s32f_AC4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image CrossCorrFull_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_16u32f_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_16u32f_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image CrossCorrFull_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevel_16u32f_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image CrossCorrFull_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

FullNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CrossCorrFull_NormLevel primitives.

1.14.28.2 CommonFullNormLevelGetBufferHostSizeParameters

Common parameters for nppiFullNormLevelGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiFullNormLevelGetBufferHostSize_8u_C1RSfs_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u_C1RSfs_Ctx`.

For common parameter descriptions, see [CommonFullNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiFullNormLevelGetBufferHostSize_8u_C3RSfs_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u_C3RSfs_Ctx`.

For common parameter descriptions, see [CommonFullNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiFullNormLevelGetBufferHostSize_8u_C4RSfs_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u_C4RSfs_Ctx`.

For common parameter descriptions, see [CommonFullNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiFullNormLevelGetBufferHostSize_8u_AC4RSfs_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u_AC4RSfs_Ctx`.

For common parameter descriptions, see [CommonFullNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiFullNormLevelGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_32f_C1R_Ctx`.

For common parameter descriptions, see [CommonFullNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiFullNormLevelGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_32f_C3R_Ctx`.

For common parameter descriptions, see [CommonFullNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiFullNormLevelGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_32f_C4R_Ctx`.

For common parameter descriptions, see [CommonFullNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiFullNormLevelGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_32f_AC4R_Ctx`.

For common parameter descriptions, see [CommonFullNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiFullNormLevelGetBufferHostSize_64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_64f_C1R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_64f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_64f_C3R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_64f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_64f_C4R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_64f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_64f_AC4R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_8u32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_8u32f_C1R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_8u32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_8u32f_C3R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_8u32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_8u32f_C4R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_8u32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_8u32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_8s32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_8s32f_C1R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_8s32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_8s32f_C3R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_8s32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_8s32f_C4R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_8s32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_8s32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_16u32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_16u32f_C1R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_16u32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_16u32f_C3R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_16u32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_16u32f_C4R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiFullNormLevelGetBufferHostSize_16u32f_AC4R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrFull_NormLevel_16u32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonFullNormLevelGetBufferHostSizeParameters*.

1.14.29. Image Cross Correlation Same Norm Level

1.14.29.1 CrossCorrSame_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with same mode.

CrossCorrSame_NormLevel

The functions compute the $\gamma_{st}(c, r)$ in *General Introduction* with same mode (see *Categorizations*).

The functions require additional scratch buffer for computations.

Note: For maximum performance oSrcRoiSize.width MUST be an integer multiple of 4.

NppStatus **nppiCrossCorrSame_NormLevel_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep,
NppiSize oSrcRoiSize, const *Npp8u*
*pTpl, int nTplStep, *NppiSize*
oTplRoiSize, *Npp8u* *pDst, int
nDstStep, int nScaleFactor, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel_8u_C3RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep,
NppiSize oSrcRoiSize, const *Npp8u*
*pTpl, int nTplStep, *NppiSize*
oTplRoiSize, *Npp8u* *pDst, int
nDstStep, int nScaleFactor, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel_8u_C4RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep,
NppiSize oSrcRoiSize, const *Npp8u*
*pTpl, int nTplStep, *NppiSize*
oTplRoiSize, *Npp8u* *pDst, int
nDstStep, int nScaleFactor, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel_8u_AC4RSfs_Ctx( const Npp8u *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp8u
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp8u *pDst, int
                                                         nDstStep, int nScaleFactor, Npp8u
                                                         *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx )
```

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel_32f_C1R_Ctx( const Npp32f *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp32f
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext nppStreamCtx )
```

One-channel 32-bit floating point image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel_32f_C3R_Ctx( const Npp32f *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp32f
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext nppStreamCtx )
```

Three-channel 32-bit floating point image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel_32f_C4R_Ctx( const Npp32f *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp32f
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext nppStreamCtx )
```

Four-channel 32-bit floating point image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel_32f_AC4R_Ctx( const Npp32f *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp32f
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext nppStreamCtx )
```

Four-channel 32-bit floating point image CrossCorrSame_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel1_64f_C1R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel1_64f_C3R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 64-bit floating point image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel1_64f_C4R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel1_64f_AC4R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrSame_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel1_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel1_8u32f_C3R_Ctx( const Npp8u *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp8u
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext nppStreamCtx )
```

Three-channel 8-bit unsigned image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel1_8u32f_C4R_Ctx( const Npp8u *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp8u
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext nppStreamCtx )
```

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel1_8u32f_AC4R_Ctx( const Npp8u *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp8u
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx )
```

Four-channel 8-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel1_8s32f_C1R_Ctx( const Npp8s *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp8s
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext nppStreamCtx )
```

One-channel 8-bit signed image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel1_8s32f_C3R_Ctx( const Npp8s *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp8s
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext nppStreamCtx )
```

Three-channel 8-bit signed image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel1_8s32f_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel1_8s32f_AC4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image CrossCorrSame_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel1_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel1_16u32f_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevel1_16u32f_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image CrossCorrSame_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevel_16u32f_AC4R_Ctx(const Npp16u *pSrc, int
                                                         nSrcStep, NppiSize oSrcRoiSize,
                                                         const Npp16u *pTpl, int
                                                         nTplStep, NppiSize oTplRoiSize,
                                                         Npp32f *pDst, int nDstStep,
                                                         Npp8u *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

Four-channel 16-bit unsigned image CrossCorrSame_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

SameNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Cross-CorrSame_NormLevel primitives.

1.14.29.2 CommonSameNormLevelGetBufferHostSizeParameters

Common parameters for nppiSameNormLevelGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

```
NppStatus nppiSameNormLevelGetBufferHostSize_8u_C1RSfs_Ctx(NppiSize oSizeROI, size_t
                                                            *hpBufferSize,
                                                            NppStreamContext
                                                            nppStreamCtx)
```

Buffer size (in bytes) for nppiCrossCorrSame_NormLevel_8u_C1RSfs.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters.*

```
NppStatus nppiSameNormLevelGetBufferHostSize_8u_C3RSfs_Ctx(NppiSize oSizeROI, size_t
                                                            *hpBufferSize,
                                                            NppStreamContext
                                                            nppStreamCtx)
```

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8u_C3RSfs_Ctx.*

For common parameter descriptions, see [CommonSameNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiSameNormLevelGetBufferHostSize_8u_C4RSfs_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8u_C4RSfs_Ctx*.

For common parameter descriptions, see [CommonSameNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8u_AC4RSfs_Ctx*.

For common parameter descriptions, see [CommonSameNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiSameNormLevelGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_32f_C1R_Ctx*.

For common parameter descriptions, see [CommonSameNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiSameNormLevelGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_32f_C3R_Ctx*.

For common parameter descriptions, see [CommonSameNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiSameNormLevelGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_32f_C4R_Ctx*.

For common parameter descriptions, see [CommonSameNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiSameNormLevelGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_32f_AC4R_Ctx*.

For common parameter descriptions, see [CommonSameNormLevelGetBufferHostSizeParameters](#).

NppStatus **nppiSameNormLevelGetBufferHostSize_64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_64f_C1R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_64f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_64f_C3R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_64f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_64f_C4R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_64f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_64f_AC4R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_8u32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8u32f_C1R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_8u32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8u32f_C3R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_8u32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8u32f_C4R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_8u32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8u32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_8s32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8s32f_C1R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_8s32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8s32f_C3R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_8s32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8s32f_C4R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_8s32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_8s32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_16u32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_16u32f_C1R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_16u32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_16u32f_C3R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_16u32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_16u32f_C4R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiSameNormLevelGetBufferHostSize_16u32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrSame_NormLevel_16u32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonSameNormLevelGetBufferHostSizeParameters*.

1.14.30. Image Cross Correlation Valid Norm Level

1.14.30.1 CrossCorrValid_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with valid mode.

CrossCorrValid_NormLevel

The functions compute the $\gamma_{st}(c, r)$ in *General Introduction* with valid mode (see *Categorizations*).

The functions require additional scratch buffer for computations.

Note: For maximum performance oSrcRoiSize.width - oTplRoiSize + 1 MUST be an integer multiple of 4.

NppStatus **nppiCrossCorrValid_NormLevel_8u_C1RSfs_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_8u_C3RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_8u_C4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_8u_AC4RSfs_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp8u* *pDst, int nDstStep, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrValid_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_64f_C1R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_64f_C3R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 64-bit floating point image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_64f_C4R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_64f_AC4R_Ctx(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrValid_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_8u32f_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_8u32f_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus nppiCrossCorrValid_NormLevel_8u32f_C4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.


```
NppStatus nppiCrossCorrValid_NormLevel1_8u32f_AC4R_Ctx(const Npp8u *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const
                                                         Npp8u *pTpl, int nTplStep,
                                                         NppiSize oTplRoiSize, Npp32f
                                                         *pDst, int nDstStep, Npp8u
                                                         *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

Four-channel 8-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_NormLevel1_8s32f_C1R_Ctx(const Npp8s *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp8s
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

One-channel 8-bit signed image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_NormLevel1_8s32f_C3R_Ctx(const Npp8s *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp8s
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

Three-channel 8-bit signed image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_NormLevel1_8s32f_C4R_Ctx(const Npp8s *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const Npp8s
                                                         *pTpl, int nTplStep, NppiSize
                                                         oTplRoiSize, Npp32f *pDst, int
                                                         nDstStep, Npp8u *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

Four-channel 8-bit signed image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_NormLevel1_8s32f_AC4R_Ctx(const Npp8s *pSrc, int nSrcStep,
                                                         NppiSize oSrcRoiSize, const
                                                         Npp8s *pTpl, int nTplStep,
                                                         NppiSize oTplRoiSize, Npp32f
                                                         *pDst, int nDstStep, Npp8u
                                                         *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx)
```


Four-channel 8-bit signed image CrossCorrValid_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_NormLevel_16u32f_C1R_Ctx(const Npp16u *pSrc, int
                                                         nSrcStep, NppiSize oSrcRoiSize,
                                                         const Npp16u *pTpl, int
                                                         nTplStep, NppiSize oTplRoiSize,
                                                         Npp32f *pDst, int nDstStep,
                                                         Npp8u *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

One-channel 16-bit unsigned image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_NormLevel_16u32f_C3R_Ctx(const Npp16u *pSrc, int
                                                         nSrcStep, NppiSize oSrcRoiSize,
                                                         const Npp16u *pTpl, int
                                                         nTplStep, NppiSize oTplRoiSize,
                                                         Npp32f *pDst, int nDstStep,
                                                         Npp8u *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

Three-channel 16-bit unsigned image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_NormLevel_16u32f_C4R_Ctx(const Npp16u *pSrc, int
                                                         nSrcStep, NppiSize oSrcRoiSize,
                                                         const Npp16u *pTpl, int
                                                         nTplStep, NppiSize oTplRoiSize,
                                                         Npp32f *pDst, int nDstStep,
                                                         Npp8u *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

Four-channel 16-bit unsigned image CrossCorrValid_NormLevel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_NormLevel_16u32f_AC4R_Ctx(const Npp16u *pSrc, int
                                                         nSrcStep, NppiSize oSrcRoiSize,
                                                         const Npp16u *pTpl, int
                                                         nTplStep, NppiSize oTplRoiSize,
                                                         Npp32f *pDst, int nDstStep,
                                                         Npp8u *pDeviceBuffer,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

Four-channel 16-bit unsigned image CrossCorrValid_NormLevel ignoring alpha channel.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

ValidNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CrossCorrValid_NormLevel primitives.

1.14.30.2 CommonValidNormLevelGetBufferHostSizeParameters

Common parameters for nppiValidNormLevelGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiValidNormLevelGetBufferHostSize_8u_C1RSfs_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u_C1RSfs_Ctx.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters.*

NppStatus **nppiValidNormLevelGetBufferHostSize_8u_C3RSfs_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for nppiCrossCorrValid_NormLevel_8u_C1RSfs_Ctx.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters.*

NppStatus **nppiValidNormLevelGetBufferHostSize_8u_C4RSfs_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_8u_C4RSfs_Ctx.*

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters.*

NppStatus **nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_8u_AC4RSfs_Ctx.*

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters.*

NppStatus **nppiValidNormLevelGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_64f_C1R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_64f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_64f_C3R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_64f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_64f_C4R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_64f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_64f_AC4R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_8u32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_8u32f_C1R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_8u32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_8u32f_C3R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_8u32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_8u32f_C4R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_8u32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_8u32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_8s32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_8s32f_C1R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_8s32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_8s32f_C3R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_8s32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_8s32f_C4R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_8s32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_8s32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_16u32f_C1R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_16u32f_C1R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_16u32f_C3R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_16u32f_C3R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_16u32f_C4R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_16u32f_C4R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

NppStatus **nppiValidNormLevelGetBufferHostSize_16u32f_AC4R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiCrossCorrValid_NormLevel_16u32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonValidNormLevelGetBufferHostSizeParameters*.

1.14.31. Image Cross Correlation Full Norm Level Advanced

1.14.31.1 CrossCorrFull_NormLevelAdvanced

Primitives for computing the normalized cross correlation coefficient between two images with full mode with large image template sizes.

CrossCorrFull_NormLevelAdvanced

The functions compute the $\gamma_{st}(c, r)$ in *General Introduction* with full mode (see *Categorizations*).

The functions require an additional scratch buffer and advanced scratch buffer for computations.

Note: For maximum performance oSrcRoiSize.width + oTplRoiSize.width - 1 MUST be an integer multiple of 4.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_64f_C1R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_64f_C3R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 64-bit floating point image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_64f_C4R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

```
NppStatus nppiCrossCorrFull_NormLevelAdvanced_8u32f_C3R_Ctx( const Npp8u *pSrc, int
                                                                nSrcStep, NppiSize
                                                                oSrcRoiSize, const
                                                                Npp8u *pTpl, int
                                                                nTplStep, NppiSize
                                                                oTplRoiSize, Npp32f
                                                                *pDst, int nDstStep,
                                                                Npp8u *pDeviceBuffer,
                                                                Npp8u *pAdvanced-
                                                                ScratchBuffer,
                                                                NppStreamContext
                                                                nppStreamCtx )
```

Three-channel 8-bit unsigned image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

```
NppStatus nppiCrossCorrFull_NormLevelAdvanced_8u32f_C4R_Ctx( const Npp8u *pSrc, int
                                                                nSrcStep, NppiSize
                                                                oSrcRoiSize, const
                                                                Npp8u *pTpl, int
                                                                nTplStep, NppiSize
                                                                oTplRoiSize, Npp32f
                                                                *pDst, int nDstStep,
                                                                Npp8u *pDeviceBuffer,
                                                                Npp8u *pAdvanced-
                                                                ScratchBuffer,
                                                                NppStreamContext
                                                                nppStreamCtx )
```

Four-channel 8-bit unsigned image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

```
NppStatus nppiCrossCorrFull_NormLevelAdvanced_8s32f_C1R_Ctx( const Npp8s *pSrc, int
                                                                nSrcStep, NppiSize
                                                                oSrcRoiSize, const Npp8s
                                                                *pTpl, int nTplStep,
                                                                NppiSize oTplRoiSize,
                                                                Npp32f *pDst, int
                                                                nDstStep, Npp8u
                                                                *pDeviceBuffer, Npp8u
                                                                *pAdvancedScratch-
                                                                Buffer,
                                                                NppStreamContext
                                                                nppStreamCtx )
```

One-channel 8-bit signed image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_8s32f_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_8s32f_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_16u32f_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrFull_NormLevelAdvanced_16u32f_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image CrossCorrFull_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

FullNormLevelGetAdvancedScratchBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CrossCorrFull_NormLevelAdvanced primitives.

NppStatus **nppiCrossCorrFull_NormLevel_GetAdvancedScratchBufferSize**(*NppiSize* oSrcRoiSize, *NppiSize* oTplRoiSize, int nSizeofDstData, int nSrcChannels, size_t *hpBufferSize)

Buffer size (in bytes) for nppiCrossCorrFull_NormLevelAdvanced functions.

Parameters

- ▶ **oSrcRoiSize** – *Region-Of-Interest (ROI)*.
- ▶ **oTplRoiSize** – *Region-Of-Interest (ROI)*.

- ▶ **nSizeofDstData** – sizeof(destination data type (usually Npp32f)).
- ▶ **nSrcChannels** – number of source image color channels.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
Scratch Buffer and Host Pointer.

Returns*Image Data Related Error Codes, ROI Related Error Codes*

1.14.32. Image Cross Correlation Same Norm Level Advanced

1.14.32.1 CrossCorrSame_NormLevelAdvanced

Primitives for computing the normalized cross correlation coefficient between two images with same mode with large image template sizes.

CrossCorrSame_NormLevelAdvanced

The functions compute the $\gamma_{st}(c, r)$ in *General Introduction* with same mode (see *Categorizations*).

The functions require an additional scratch buffer and advanced scratch buffer for computations.

Note: For maximum performance oSrcRoiSize.width MUST be an integer multiple of 4.

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus nppiCrossCorrSame_NormLevelAdvanced_32f_C4R_Ctx(const [Npp32f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer, [Npp8u](#) *pAdvancedScratchBuffer, [NppStreamContext](#) nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus nppiCrossCorrSame_NormLevelAdvanced_64f_C1R_Ctx(const [Npp64f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp64f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp64f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer, [Npp8u](#) *pAdvancedScratchBuffer, [NppStreamContext](#) nppStreamCtx)

One-channel 64-bit floating point image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus nppiCrossCorrSame_NormLevelAdvanced_64f_C3R_Ctx(const [Npp64f](#) *pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp64f](#) *pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp64f](#) *pDst, int nDstStep, [Npp8u](#) *pDeviceBuffer, [Npp8u](#) *pAdvancedScratchBuffer, [NppStreamContext](#) nppStreamCtx)

Three-channel 64-bit floating point image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see [Common parameters for nppiSqrDistanceFull functions include:](#).

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_64f_C4R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_8u32f_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_8u32f_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_8s32f_C1R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_8s32f_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_8s32f_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrSame_NormLevelAdvanced_16u32f_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrSame_NormLevelAdvanced_16u32f_C4R_Ctx( const Npp16u *pSrc, int
                                                                nSrcStep, NppiSize
                                                                oSrcRoiSize, const
                                                                Npp16u *pTpl, int
                                                                nTplStep, NppiSize
                                                                oTplRoiSize, Npp32f
                                                                *pDst, int nDstStep,
                                                                Npp8u *pDeviceBuffer,
                                                                Npp8u *pAdvanced-
                                                                ScratchBuffer,
                                                                NppStreamContext
                                                                nppStreamCtx )
```

Four-channel 16-bit unsigned image CrossCorrSame_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

SameNormLevelGetAdvancedScratchBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Cross-CorrSame_NormLevelAdvanced primitives.

```
NppStatus nppiCrossCorrSame_NormLevel1_GetAdvancedScratchBufferSize( NppiSize
                                                                oSrcRoiSize,
                                                                NppiSize
                                                                oTplRoiSize, int
                                                                nSizeofDstData,
                                                                int
                                                                nSrcChannels,
                                                                size_t
                                                                *hpBufferSize )
```

Buffer size (in bytes) for nppiCrossCorrSame_NormLevelAdvanced functions.

Parameters

- ▶ **oSrcRoiSize** – *Region-Of-Interest (ROI)*.
- ▶ **oTplRoiSize** – *Region-Of-Interest (ROI)*.
- ▶ **nSizeofDstData** – sizeof(destination data type (usually Npp32f)).
- ▶ **nSrcChannels** – number of source image color channels.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.14.33. Image Cross Correlation Valid Norm Level Advanced

1.14.33.1 CrossCorrValid_NormLevelAdvanced

Primitives for computing the normalized cross correlation coefficient between two images with valid mode with large template sizes.

CrossCorrValid_NormLevelAdvanced

The functions compute the $\gamma_{st}(c, r)$ in *General Introduction* with valid mode (see *Categorizations*).

The functions require an additional scratch buffer and advanced scratch buffer for computations.

Note: For maximum performance oSrcRoiSize.width - oTplRoiSize + 1 MUST be an integer multiple of 4.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_32f_C4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp32f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_64f_C1R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_64f_C3R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 64-bit floating point image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_64f_C4R_Ctx**(const *Npp64f* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp64f* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp64f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_8u32f_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_8u32f_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvancedScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_8u32f_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvanced-ScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_8s32f_C1R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvanced-ScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_8s32f_C3R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvanced-ScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_8s32f_C4R_Ctx**(const *Npp8s* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp8s* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvanced-ScratchBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_16u32f_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvanced-ScratchBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

NppStatus **nppiCrossCorrValid_NormLevelAdvanced_16u32f_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSrcRoiSize, const *Npp16u* *pTpl, int nTplStep, *NppiSize* oTplRoiSize, *Npp32f* *pDst, int nDstStep, *Npp8u* *pDeviceBuffer, *Npp8u* *pAdvanced-ScratchBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

```
NppStatus nppiCrossCorrValid_NormLevelAdvanced_16u32f_C4R_Ctx(const Npp16u *pSrc,
                                                                int nSrcStep, NppiSize
                                                                oSrcRoiSize, const
                                                                Npp16u *pTpl, int
                                                                nTplStep, NppiSize
                                                                oTplRoiSize, Npp32f
                                                                *pDst, int nDstStep,
                                                                Npp8u *pDeviceBuffer,
                                                                Npp8u *pAdvanced-
                                                                ScratchBuffer,
                                                                NppStreamContext
                                                                nppStreamCtx)
```

Four-channel 16-bit unsigned image CrossCorrValid_NormLevelAdvanced.

For common parameter descriptions, see *Common parameters for nppiSqrDistanceFull functions include:*.

ValidNormLevelGetAdvancedScratchBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CrossCorrValid_NormLevelAdvanced primitives.

```
NppStatus nppiCrossCorrValid_NormLevel1_GetAdvancedScratchBufferSize(NppiSize
                                                                        oSrcRoiSize,
                                                                        NppiSize
                                                                        oTplRoiSize,
                                                                        int nSizeofDst-
                                                                        Data, int
                                                                        nSrcChannels,
                                                                        size_t
                                                                        *hpBufferSize)
```

Buffer size (in bytes) for nppiCrossCorrValid_NormLevelAdvanced functions.

Parameters

- ▶ **oSrcRoiSize** – *Region-Of-Interest (ROI)*.
- ▶ **oTplRoiSize** – *Region-Of-Interest (ROI)*.
- ▶ **nSizeofDstData** – sizeof(destination data type (usually Npp32f)).
- ▶ **nSrcChannels** – number of source image color channels.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.14.34. Image Quality Index

1.14.34.1 Image Quality Index

Primitives for computing the image quality index of two images.

QualityIndex

Given two images M and N (both $W \times H$), the mathematical formula to calculate the image quality index Q between them is expressed as:

$$Q = \frac{4\sigma_{MN}\tilde{M}\tilde{N}}{[(\tilde{M}^2) + (\tilde{N}^2)][(\sigma_M)^2 + (\sigma_N)^2]}$$

where

$$\tilde{M} = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} M(j, i)$$

$$\tilde{N} = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} N(j, i)$$

$$\sigma_M = \sqrt{\frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [M(j, i) - \tilde{M}]^2}$$

$$\sigma_N = \sqrt{\frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [N(j, i) - \tilde{N}]^2}$$

$$\sigma_{MN} = \frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [M(j, i) - \tilde{M}][N(j, i) - \tilde{N}]$$

The functions require additional scratch buffer for computations.

1.14.34.2 Common parameters for nppiQualityIndex functions include:

param pSrc1

Source-Image Pointer.

param nSrc1Step

Source-Image Line Step.

param pSrc2

Source-Image Pointer.

param nSrc2Step

Source-Image Line Step.

param oRoiSize

Region-Of-Interest (ROI).

param pDst

Pointer to the quality index.

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, *ROI Related Error Codes*, or
NPP_QUALITY_INDEX_ERROR if pixels of either image are constant numberse.

NppStatus **nppiQualityIndex_8u32f_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oRoiSize, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image QualityIndex.

For common parameter descriptions, see *Common parameters for nppiQualityIndex functions include:*.

NppStatus **nppiQualityIndex_16u32f_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oRoiSize, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image QualityIndex.

For common parameter descriptions, see *Common parameters for nppiQualityIndex functions include:*.

NppStatus **nppiQualityIndex_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oRoiSize, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image QualityIndex.

For common parameter descriptions, see *Common parameters for nppiQualityIndex functions include:*.

NppStatus **nppiQualityIndex_8u32f_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oRoiSize, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image QualityIndex.

For common parameter descriptions, see *Common parameters for nppiQualityIndex functions include:*.

NppStatus **nppiQualityIndex_16u32f_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oRoiSize, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image QualityIndex.

For common parameter descriptions, see *Common parameters for nppiQualityIndex functions include:*.

NppStatus **nppiQualityIndex_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oRoiSize, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit floating point image QualityIndex.

For common parameter descriptions, see *Common parameters for nppiQualityIndex functions include:*.

NppStatus **nppiQualityIndex_8u32f_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oRoiSize, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image QualityIndex.

For common parameter descriptions, see *Common parameters for nppiQualityIndex functions include:*.

NppStatus **nppiQualityIndex_16u32f_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oRoiSize, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image QualityIndex.

For common parameter descriptions, see *Common parameters for nppiQualityIndex functions include:*.

NppStatus **nppiQualityIndex_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oRoiSize, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image QualityIndex.

For common parameter descriptions, see *Common parameters for nppiQualityIndex functions include:*.

QualityIndexGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the QualityIndex primitives.

1.14.34.3 CommonQualityIndexGetBufferHostSizeParameters

Common parameters for nppiQualityIndexGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes*.

NppStatus **nppiQualityIndexGetBufferHostSize_8u32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiQualityIndex_8u32f_C1R_Ctx*.

For common parameter descriptions, see *CommonQualityIndexGetBufferHostSizeParameters*.

NppStatus **nppiQualityIndexGetBufferHostSize_16u32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiQualityIndex_16u32f_C1R_Ctx*.

For common parameter descriptions, see *CommonQualityIndexGetBufferHostSizeParameters*.

NppStatus **nppiQualityIndexGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiQualityIndex_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonQualityIndexGetBufferHostSizeParameters*.

NppStatus **nppiQualityIndexGetBufferHostSize_8u32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiQualityIndex_8u32f_C3R_Ctx*.

For common parameter descriptions, see *CommonQualityIndexGetBufferHostSizeParameters*.

NppStatus **nppiQualityIndexGetBufferHostSize_16u32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiQualityIndex_16u32f_C3R_Ctx*.

For common parameter descriptions, see *CommonQualityIndexGetBufferHostSizeParameters*.

NppStatus **nppiQualityIndexGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiQualityIndex_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonQualityIndexGetBufferHostSizeParameters*.

NppStatus **nppiQualityIndexGetBufferHostSize_8u32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size (in bytes) for *nppiQualityIndex_8u32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonQualityIndexGetBufferHostSizeParameters*.

NppStatus **nppiQualityIndexGetBufferHostSize_16u32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiQualityIndex_16u32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonQualityIndexGetBufferHostSizeParameters*.

NppStatus **nppiQualityIndexGetBufferHostSize_32f_AC4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size (in bytes) for *nppiQualityIndex_32f_AC4R_Ctx*.

For common parameter descriptions, see *CommonQualityIndexGetBufferHostSizeParameters*.

1.14.35. Image Maximum Error

1.14.35.1 MaximumError

Primitives for computing the maximum error between two images. Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*, the maximum error is defined as the largest absolute difference between pixels of two images. If the image is in complex format, the absolute value of the complex number is provided.

1.14.35.2 Common parameters for nppiMaximumError functions include:

param pSrc1

Source-Image Pointer.

param nSrc1Step

Source-Image Line Step.

param pSrc2

Source-Image Pointer.

param nSrc2Step

Source-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param pError

Device memory pointer to the computed error.

param pDeviceBuffer

Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes.

MaximumError

NppStatus **nppiMaximumError_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiMaximumError_8s_C1R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiMaximumError_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiMaximumError_16s_C1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiMaximumError_16sc_C1R_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed complex image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiMaximumError_32u_C1R_Ctx**(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_32s_C1R_Ctx(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_32sc_C1R_Ctx(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit signed complex image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_32f_C1R_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_32fc_C1R_Ctx(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point complex image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_64f_C1R_Ctx(const *Npp64f* *pSrc1, int nSrc1Step, const *Npp64f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_8u_C2R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 8-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_8s_C2R_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 8-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_16u_C2R_Ctx(const Npp16u *pSrc1, int nSrc1Step, const Npp16u
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Two-channel 16-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_16s_C2R_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Two-channel 16-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_16sc_C2R_Ctx(const Npp16sc *pSrc1, int nSrc1Step, const
                                           Npp16sc *pSrc2, int nSrc2Step, NppiSize oSizeROI,
                                           Npp64f *pError, Npp8u *pDeviceBuffer,
                                           NppStreamContext nppStreamCtx)
```

Two-channel 16-bit signed complex image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_32u_C2R_Ctx(const Npp32u *pSrc1, int nSrc1Step, const Npp32u
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Two-channel 32-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_32s_C2R_Ctx(const Npp32s *pSrc1, int nSrc1Step, const Npp32s
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Two-channel 32-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_32sc_C2R_Ctx(const Npp32sc *pSrc1, int nSrc1Step, const
                                           Npp32sc *pSrc2, int nSrc2Step, NppiSize oSizeROI,
                                           Npp64f *pError, Npp8u *pDeviceBuffer,
                                           NppStreamContext nppStreamCtx)
```

Two-channel 32-bit signed complex image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_32f_C2R_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 32-bit floating point image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_32fc_C2R_Ctx(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 32-bit floating point complex image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_64f_C2R_Ctx(const *Npp64f* *pSrc1, int nSrc1Step, const *Npp64f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 64-bit floating point image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_8u_C3R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_8s_C3R_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_16u_C3R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_16s_C3R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_16sc_C3R_Ctx(const Npp16sc *pSrc1, int nSrc1Step, const  
                                           Npp16sc *pSrc2, int nSrc2Step, NppiSize oSizeROI,  
                                           Npp64f *pError, Npp8u *pDeviceBuffer,  
                                           NppStreamContext nppStreamCtx)
```

Three-channel 16-bit signed complex image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_32u_C3R_Ctx(const Npp32u *pSrc1, int nSrc1Step, const Npp32u  
                                           *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f  
                                           *pError, Npp8u *pDeviceBuffer, NppStreamContext  
                                           nppStreamCtx)
```

Three-channel 32-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_32s_C3R_Ctx(const Npp32s *pSrc1, int nSrc1Step, const Npp32s  
                                           *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f  
                                           *pError, Npp8u *pDeviceBuffer, NppStreamContext  
                                           nppStreamCtx)
```

Three-channel 32-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_32sc_C3R_Ctx(const Npp32sc *pSrc1, int nSrc1Step, const  
                                           Npp32sc *pSrc2, int nSrc2Step, NppiSize oSizeROI,  
                                           Npp64f *pError, Npp8u *pDeviceBuffer,  
                                           NppStreamContext nppStreamCtx)
```

Three-channel 32-bit signed complex image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_32f_C3R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f  
                                           *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f  
                                           *pError, Npp8u *pDeviceBuffer, NppStreamContext  
                                           nppStreamCtx)
```

Three-channel 32-bit floating point image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumError_32fc_C3R_Ctx(const Npp32fc *pSrc1, int nSrc1Step, const  
                                           Npp32fc *pSrc2, int nSrc2Step, NppiSize oSizeROI,  
                                           Npp64f *pError, Npp8u *pDeviceBuffer,  
                                           NppStreamContext nppStreamCtx)
```

Three-channel 32-bit floating point complex image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_64f_C3R_Ctx(const *Npp64f* *pSrc1, int nSrc1Step, const *Npp64f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 64-bit floating point image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_8u_C4R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_8s_C4R_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_16u_C4R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_16s_C4R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_16sc_C4R_Ctx(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed complex image Maximum_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumError_32u_C4R_Ctx(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit unsigned image Maximum_Error.

For common parameter descriptions, see [Common parameters for nppiMaximumError functions include:](#).

```
NppStatus nppiMaximumError_32s_C4R_Ctx(const Npp32s *pSrc1, int nSrc1Step, const Npp32s
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Four-channel 32-bit signed image Maximum_Error.

For common parameter descriptions, see [Common parameters for nppiMaximumError functions include:](#).

```
NppStatus nppiMaximumError_32sc_C4R_Ctx(const Npp32sc *pSrc1, int nSrc1Step, const
                                           Npp32sc *pSrc2, int nSrc2Step, NppiSize oSizeROI,
                                           Npp64f *pError, Npp8u *pDeviceBuffer,
                                           NppStreamContext nppStreamCtx)
```

Four-channel 32-bit signed complex image Maximum_Error.

For common parameter descriptions, see [Common parameters for nppiMaximumError functions include:](#).

```
NppStatus nppiMaximumError_32f_C4R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Four-channel 32-bit floating point image Maximum_Error.

For common parameter descriptions, see [Common parameters for nppiMaximumError functions include:](#).

```
NppStatus nppiMaximumError_32fc_C4R_Ctx(const Npp32fc *pSrc1, int nSrc1Step, const
                                           Npp32fc *pSrc2, int nSrc2Step, NppiSize oSizeROI,
                                           Npp64f *pError, Npp8u *pDeviceBuffer,
                                           NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating point complex image Maximum_Error.

For common parameter descriptions, see [Common parameters for nppiMaximumError functions include:](#).

```
NppStatus nppiMaximumError_64f_C4R_Ctx(const Npp64f *pSrc1, int nSrc1Step, const Npp64f
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Four-channel 64-bit floating point image Maximum_Error.

For common parameter descriptions, see [Common parameters for nppiMaximumError functions include:](#).

MaximumErrorGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MaximumError primitives.

1.14.35.3 CommonMaximumErrorGetBufferHostSizeParameters

Common parameters for nppiMaximumErrorGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus **nppiMaximumErrorGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_8s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_8s_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16sc_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_16sc_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32u_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32s_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32s_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32sc_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32sc_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32fc_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32fc_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_64f_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_8u_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_8u_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_8s_C2R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_8s_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16u_C2R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_16u_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16s_C2R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_16s_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16sc_C2R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_16sc_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32u_C2R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_32u_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32s_C2R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_32s_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32sc_C2R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_32sc_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32f_C2R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_32f_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32fc_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32fc_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_64f_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_64f_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_8s_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_8s_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16sc_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_16sc_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32u_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32s_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32s_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32sc_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32sc_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32fc_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32fc_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_64f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_64f_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_8s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_8s_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_16sc_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_16sc_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32u_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32s_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32sc_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32sc_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumError_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_32fc_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_32fc_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumErrorGetBufferHostSize_64f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMaximumError_64f_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumErrorGetBufferHostSizeParameters*.

1.14.36. Image Average Error

1.14.36.1 AverageError

Primitives for computing the average error between two images. Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*, the average error is defined as:

$$AverageError = \frac{1}{W \cdot H \cdot N} \sum_{n=0}^{N-1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|$$

where *N* stands for the number of channels. If the image is in complex format, the absolute value is used for computation.

AverageError

NppStatus **nppiAverageError_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiAverageError_8s_C1R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiAverageError_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_16s_C1R_Ctx(const Npp16s *pSrc1, int nSrc1Step, const Npp16s
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

One-channel 16-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_16sc_C1R_Ctx(const Npp16sc *pSrc1, int nSrc1Step, const
                                           Npp16sc *pSrc2, int nSrc2Step, NppiSize oSizeROI,
                                           Npp64f *pError, Npp8u *pDeviceBuffer,
                                           NppStreamContext nppStreamCtx)
```

One-channel 16-bit signed complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32u_C1R_Ctx(const Npp32u *pSrc1, int nSrc1Step, const Npp32u
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

One-channel 32-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32s_C1R_Ctx(const Npp32s *pSrc1, int nSrc1Step, const Npp32s
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

One-channel 32-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32sc_C1R_Ctx(const Npp32sc *pSrc1, int nSrc1Step, const
                                           Npp32sc *pSrc2, int nSrc2Step, NppiSize oSizeROI,
                                           Npp64f *pError, Npp8u *pDeviceBuffer,
                                           NppStreamContext nppStreamCtx)
```

One-channel 32-bit signed complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32f_C1R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

One-channel 32-bit floating point image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_32fc_C1R_Ctx(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit floating point complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_64f_C1R_Ctx(const *Npp64f* *pSrc1, int nSrc1Step, const *Npp64f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 64-bit floating point image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_8u_C2R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 8-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_8s_C2R_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 8-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_16u_C2R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 16-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_16s_C2R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 16-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_16sc_C2R_Ctx(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 16-bit signed complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32u_C2R_Ctx(const Npp32u *pSrc1, int nSrc1Step, const Npp32u
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Two-channel 32-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32s_C2R_Ctx(const Npp32s *pSrc1, int nSrc1Step, const Npp32s
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Two-channel 32-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32sc_C2R_Ctx(const Npp32sc *pSrc1, int nSrc1Step, const
                                         Npp32sc *pSrc2, int nSrc2Step, NppiSize oSizeROI,
                                         Npp64f *pError, Npp8u *pDeviceBuffer,
                                         NppStreamContext nppStreamCtx)
```

Two-channel 32-bit signed complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32f_C2R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Two-channel 32-bit floating point image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32fc_C2R_Ctx(const Npp32fc *pSrc1, int nSrc1Step, const
                                         Npp32fc *pSrc2, int nSrc2Step, NppiSize oSizeROI,
                                         Npp64f *pError, Npp8u *pDeviceBuffer,
                                         NppStreamContext nppStreamCtx)
```

Two-channel 32-bit floating point complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_64f_C2R_Ctx(const Npp64f *pSrc1, int nSrc1Step, const Npp64f
                                         *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                         *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                         nppStreamCtx)
```

Two-channel 64-bit floating point image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_8u_C3R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_8s_C3R_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_16u_C3R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_16s_C3R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_16sc_C3R_Ctx(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit signed complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_32u_C3R_Ctx(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_32s_C3R_Ctx(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 32-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32sc_C3R_Ctx(const Npp32sc *pSrc1, int nSrc1Step, const  
                                           Npp32sc *pSrc2, int nSrc2Step, NppiSize oSizeROI,  
                                           Npp64f *pError, Npp8u *pDeviceBuffer,  
                                           NppStreamContext nppStreamCtx)
```

Three-channel 32-bit signed complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32f_C3R_Ctx(const Npp32f *pSrc1, int nSrc1Step, const Npp32f  
                                           *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f  
                                           *pError, Npp8u *pDeviceBuffer, NppStreamContext  
                                           nppStreamCtx)
```

Three-channel 32-bit floating point image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32fc_C3R_Ctx(const Npp32fc *pSrc1, int nSrc1Step, const  
                                           Npp32fc *pSrc2, int nSrc2Step, NppiSize oSizeROI,  
                                           Npp64f *pError, Npp8u *pDeviceBuffer,  
                                           NppStreamContext nppStreamCtx)
```

Three-channel 32-bit floating point complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_64f_C3R_Ctx(const Npp64f *pSrc1, int nSrc1Step, const Npp64f  
                                           *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f  
                                           *pError, Npp8u *pDeviceBuffer, NppStreamContext  
                                           nppStreamCtx)
```

Three-channel 64-bit floating point image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_8u_C4R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const Npp8u  
                                           *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f  
                                           *pError, Npp8u *pDeviceBuffer, NppStreamContext  
                                           nppStreamCtx)
```

Four-channel 8-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_8s_C4R_Ctx(const Npp8s *pSrc1, int nSrc1Step, const Npp8s  
                                           *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f  
                                           *pError, Npp8u *pDeviceBuffer, NppStreamContext  
                                           nppStreamCtx)
```

Four-channel 8-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_16u_C4R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_16s_C4R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_16sc_C4R_Ctx(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_32u_C4R_Ctx(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit unsigned image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_32s_C4R_Ctx(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_32sc_C4R_Ctx(const *Npp32sc* *pSrc1, int nSrc1Step, const *Npp32sc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageError_32f_C4R_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_32fc_C4R_Ctx( const Npp32fc *pSrc1, int nSrc1Step, const
                                           Npp32fc *pSrc2, int nSrc2Step, NppiSize oSizeROI,
                                           Npp64f *pError, Npp8u *pDeviceBuffer,
                                           NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating point complex image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageError_64f_C4R_Ctx( const Npp64f *pSrc1, int nSrc1Step, const Npp64f
                                           *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f
                                           *pError, Npp8u *pDeviceBuffer, NppStreamContext
                                           nppStreamCtx)
```

Four-channel 64-bit floating point image Average_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

AverageErrorGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

1.14.36.2 CommonAverageErrorGetBufferHostSizeParameters

Common parameters for nppiSumAverageErrorBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

```
NppStatus nppiAverageErrorGetBufferHostSize_8u_C1R_Ctx( NppiSize oSizeROI, size_t
                                                         *hpBufferSize,
                                                         NppStreamContext
                                                         nppStreamCtx)
```

Buffer size for *nppiAverageError_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters.*

```
NppStatus nppiAverageErrorGetBufferHostSize_8s_C1R_Ctx( NppiSize oSizeROI, size_t
                                                         *hpBufferSize,
                                                         NppStreamContext
                                                         nppStreamCtx)
```


Buffer size for *nppiAverageError_8s_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16sc_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_16sc_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32u_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32s_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32s_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32sc_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32sc_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32fc_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32fc_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_64f_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_64f_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_8u_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_8u_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_8s_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_8s_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16u_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_16u_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16s_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_16s_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16sc_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_16sc_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32u_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32u_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32s_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32s_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32sc_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32sc_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32f_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32f_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32fc_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32fc_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_64f_C2R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_64f_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_8s_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_8s_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16sc_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_16sc_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_32u_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32s_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_32s_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32sc_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_32sc_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32fc_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_32fc_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_64f_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_64f_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_8s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_8s_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_16sc_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_16sc_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageError_32u_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32s_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_32s_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32sc_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_32sc_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_32fc_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_32fc_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageErrorGetBufferHostSize_64f_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageError_64f_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageErrorGetBufferHostSizeParameters*.

1.14.37. Image Maximum Relative Error

1.14.37.1 MaximumRelativeError

Primitives for computing the maximum relative error between two images. Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*, the maximum relative error is defined as:

$$\text{MaximumRelativeError} = \max \frac{|pSrc1(j, i) - pSrc2(j, i)|}{\max(|pSrc1(j, i)|, |pSrc2(j, i)|)}$$

If the image is in complex format, the absolute value is used for computation. For multiple channels, the maximum relative error of all the channels is returned.

MaximumRelativeError

NppStatus **nppiMaximumRelativeError_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiMaximumRelativeError_8s_C1R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiMaximumRelativeError_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiMaximumRelativeError_16s_C1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiMaximumRelativeError_16sc_C1R_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 16-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiMaximumRelativeError_32u_C1R_Ctx**(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 32-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32s_C1R_Ctx(const Npp32s *pSrc1, int nSrc1Step,  
                                                const Npp32s *pSrc2, int nSrc2Step,  
                                                NppiSize oSizeROI, Npp64f *pError, Npp8u  
                                                *pDeviceBuffer, NppStreamContext  
                                                nppStreamCtx)
```

One-channel 32-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32sc_C1R_Ctx(const Npp32sc *pSrc1, int nSrc1Step,  
                                                const Npp32sc *pSrc2, int nSrc2Step,  
                                                NppiSize oSizeROI, Npp64f *pError,  
                                                Npp8u *pDeviceBuffer,  
                                                NppStreamContext nppStreamCtx)
```

One-channel 32-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32f_C1R_Ctx(const Npp32f *pSrc1, int nSrc1Step,  
                                                const Npp32f *pSrc2, int nSrc2Step,  
                                                NppiSize oSizeROI, Npp64f *pError, Npp8u  
                                                *pDeviceBuffer, NppStreamContext  
                                                nppStreamCtx)
```

One-channel 32-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32fc_C1R_Ctx(const Npp32fc *pSrc1, int nSrc1Step,  
                                                const Npp32fc *pSrc2, int nSrc2Step,  
                                                NppiSize oSizeROI, Npp64f *pError,  
                                                Npp8u *pDeviceBuffer,  
                                                NppStreamContext nppStreamCtx)
```

One-channel 32-bit floating point complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_64f_C1R_Ctx(const Npp64f *pSrc1, int nSrc1Step,  
                                                const Npp64f *pSrc2, int nSrc2Step,  
                                                NppiSize oSizeROI, Npp64f *pError, Npp8u  
                                                *pDeviceBuffer, NppStreamContext  
                                                nppStreamCtx)
```

One-channel 64-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_8u_C2R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 8-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_8s_C2R_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 8-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_16u_C2R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 16-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_16s_C2R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 16-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_16sc_C2R_Ctx(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 16-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_32u_C2R_Ctx(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 32-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32s_C2R_Ctx(const Npp32s *pSrc1, int nSrc1Step,  
const Npp32s *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError, Npp8u  
*pDeviceBuffer, NppStreamContext  
nppStreamCtx)
```

Two-channel 32-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32sc_C2R_Ctx(const Npp32sc *pSrc1, int nSrc1Step,  
const Npp32sc *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError,  
Npp8u *pDeviceBuffer,  
NppStreamContext nppStreamCtx)
```

Two-channel 32-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32f_C2R_Ctx(const Npp32f *pSrc1, int nSrc1Step,  
const Npp32f *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError, Npp8u  
*pDeviceBuffer, NppStreamContext  
nppStreamCtx)
```

Two-channel 32-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32fc_C2R_Ctx(const Npp32fc *pSrc1, int nSrc1Step,  
const Npp32fc *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError,  
Npp8u *pDeviceBuffer,  
NppStreamContext nppStreamCtx)
```

Two-channel 32-bit floating point complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_64f_C2R_Ctx(const Npp64f *pSrc1, int nSrc1Step,  
const Npp64f *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError, Npp8u  
*pDeviceBuffer, NppStreamContext  
nppStreamCtx)
```

Two-channel 64-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_8u_C3R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const  
Npp8u *pSrc2, int nSrc2Step, NppiSize  
oSizeROI, Npp64f *pError, Npp8u  
*pDeviceBuffer, NppStreamContext  
nppStreamCtx)
```

Three-channel 8-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_8s_C3R_Ctx(const Npp8s *pSrc1, int nSrc1Step, const
                                                Npp8s *pSrc2, int nSrc2Step, NppiSize
                                                oSizeROI, Npp64f *pError, Npp8u
                                                *pDeviceBuffer, NppStreamContext
                                                nppStreamCtx)
```

Three-channel 8-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_16u_C3R_Ctx(const Npp16u *pSrc1, int nSrc1Step,
                                                const Npp16u *pSrc2, int nSrc2Step,
                                                NppiSize oSizeROI, Npp64f *pError, Npp8u
                                                *pDeviceBuffer, NppStreamContext
                                                nppStreamCtx)
```

Three-channel 16-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_16s_C3R_Ctx(const Npp16s *pSrc1, int nSrc1Step,
                                                const Npp16s *pSrc2, int nSrc2Step,
                                                NppiSize oSizeROI, Npp64f *pError, Npp8u
                                                *pDeviceBuffer, NppStreamContext
                                                nppStreamCtx)
```

Three-channel 16-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_16sc_C3R_Ctx(const Npp16sc *pSrc1, int nSrc1Step,
                                                const Npp16sc *pSrc2, int nSrc2Step,
                                                NppiSize oSizeROI, Npp64f *pError,
                                                Npp8u *pDeviceBuffer,
                                                NppStreamContext nppStreamCtx)
```

Three-channel 16-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32u_C3R_Ctx(const Npp32u *pSrc1, int nSrc1Step,
                                                const Npp32u *pSrc2, int nSrc2Step,
                                                NppiSize oSizeROI, Npp64f *pError, Npp8u
                                                *pDeviceBuffer, NppStreamContext
                                                nppStreamCtx)
```

Three-channel 32-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32s_C3R_Ctx(const Npp32s *pSrc1, int nSrc1Step,
const Npp32s *pSrc2, int nSrc2Step,
NppiSize oSizeROI, Npp64f *pError, Npp8u
*pDeviceBuffer, NppStreamContext
nppStreamCtx)
```

Three-channel 32-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32sc_C3R_Ctx(const Npp32sc *pSrc1, int nSrc1Step,
const Npp32sc *pSrc2, int nSrc2Step,
NppiSize oSizeROI, Npp64f *pError,
Npp8u *pDeviceBuffer,
NppStreamContext nppStreamCtx)
```

Three-channel 32-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32f_C3R_Ctx(const Npp32f *pSrc1, int nSrc1Step,
const Npp32f *pSrc2, int nSrc2Step,
NppiSize oSizeROI, Npp64f *pError, Npp8u
*pDeviceBuffer, NppStreamContext
nppStreamCtx)
```

Three-channel 32-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32fc_C3R_Ctx(const Npp32fc *pSrc1, int nSrc1Step,
const Npp32fc *pSrc2, int nSrc2Step,
NppiSize oSizeROI, Npp64f *pError,
Npp8u *pDeviceBuffer,
NppStreamContext nppStreamCtx)
```

Three-channel 32-bit floating point complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_64f_C3R_Ctx(const Npp64f *pSrc1, int nSrc1Step,
const Npp64f *pSrc2, int nSrc2Step,
NppiSize oSizeROI, Npp64f *pError, Npp8u
*pDeviceBuffer, NppStreamContext
nppStreamCtx)
```

Three-channel 64-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_8u_C4R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const
Npp8u *pSrc2, int nSrc2Step, NppiSize
oSizeROI, Npp64f *pError, Npp8u
*pDeviceBuffer, NppStreamContext
nppStreamCtx)
```

Four-channel 8-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_8s_C4R_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_16u_C4R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_16s_C4R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step, const *Npp16s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_16sc_C4R_Ctx(const *Npp16sc* *pSrc1, int nSrc1Step, const *Npp16sc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 16-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_32u_C4R_Ctx(const *Npp32u* *pSrc1, int nSrc1Step, const *Npp32u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiMaximumRelativeError_32s_C4R_Ctx(const *Npp32s* *pSrc1, int nSrc1Step, const *Npp32s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32sc_C4R_Ctx(const Npp32sc *pSrc1, int nSrc1Step,  
const Npp32sc *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError,  
Npp8u *pDeviceBuffer,  
NppStreamContext nppStreamCtx)
```

Four-channel 32-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32f_C4R_Ctx(const Npp32f *pSrc1, int nSrc1Step,  
const Npp32f *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError, Npp8u  
*pDeviceBuffer, NppStreamContext  
nppStreamCtx)
```

Four-channel 32-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_32fc_C4R_Ctx(const Npp32fc *pSrc1, int nSrc1Step,  
const Npp32fc *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError,  
Npp8u *pDeviceBuffer,  
NppStreamContext nppStreamCtx)
```

Four-channel 32-bit floating point complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiMaximumRelativeError_64f_C4R_Ctx(const Npp64f *pSrc1, int nSrc1Step,  
const Npp64f *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError, Npp8u  
*pDeviceBuffer, NppStreamContext  
nppStreamCtx)
```

Four-channel 64-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

MaximumRelativeErrorGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

1.14.37.2 CommonMaximumRelativeErrorGetBufferHostSizeParameters

Common parameters for `nppiMaximumRelativeErrorGetBufferHostSize` functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: `hpBufferSize` is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

`NPP_NULL_POINTER_ERROR` if `hpBufferSize` is 0 (NULL), *ROI Related Error Codes.*

NppStatus `nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R_Ctx`(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus `nppiMaximumRelativeErrorGetBufferHostSize_8s_C1R_Ctx`(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_8s_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus `nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R_Ctx`(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus `nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R_Ctx`(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_16sc_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16sc_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32u_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32u_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32s_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32s_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32sc_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32sc_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32fc_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32fc_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_64f_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_64f_C1R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_8u_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_8s_C2R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_8s_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16u_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16s_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_16sc_C2R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16sc_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32u_C2R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32u_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32s_C2R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32s_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32sc_C2R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32sc_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32f_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32fc_C2R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32fc_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_64f_C2R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_64f_C2R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_8s_C3R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_8s_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_16sc_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16sc_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32u_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32u_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32s_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32s_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32sc_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32sc_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32fc_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32fc_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_64f_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_64f_C3R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_8s_C4R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_8s_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16u_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16s_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_16sc_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_16sc_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32u_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32u_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32s_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32s_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32sc_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32sc_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_32fc_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_32fc_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiMaximumRelativeErrorGetBufferHostSize_64f_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiMaximumRelativeError_64f_C4R_Ctx*.

For common parameter descriptions, see *CommonMaximumRelativeErrorGetBufferHostSizeParameters*.

1.14.38. Image Average Relative Error

1.14.38.1 AverageRelativeError

Primitives for computing the average relative error between two images. Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*, the maximum relative error is defined as:

$$AverageRelativeError = \frac{1}{W \cdot H \cdot N} \sum_{n=0}^{N-1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} \frac{|pSrc1(j, i) - pSrc2(j, i)|}{\max(|pSrc1(j, i)|, |pSrc2(j, i)|)}$$

where *N* is the number of channels. If the image is in complex format, the absolute value is used for computation.

AverageRelativeError

NppStatus **nppiAverageRelativeError_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const
Npp8u *pSrc2, int nSrc2Step, *NppiSize*
oSizeROI, *Npp64f* *pError, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

One-channel 8-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiAverageRelativeError_8s_C1R_Ctx**(const *Npp8s* *pSrc1, int nSrc1Step, const
Npp8s *pSrc2, int nSrc2Step, *NppiSize*
oSizeROI, *Npp64f* *pError, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

One-channel 8-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiAverageRelativeError_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step,
const *Npp16u* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

One-channel 16-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiAverageRelativeError_16s_C1R_Ctx**(const *Npp16s* *pSrc1, int nSrc1Step,
const *Npp16s* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

One-channel 16-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiAverageRelativeError_16sc_C1R_Ctx**(const *Npp16sc* *pSrc1, int nSrc1Step,
const *Npp16sc* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError,
Npp8u *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 16-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiAverageRelativeError_32u_C1R_Ctx**(const *Npp32u* *pSrc1, int nSrc1Step,
const *Npp32u* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

One-channel 32-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiAverageRelativeError_32s_C1R_Ctx**(const *Npp32s* *pSrc1, int nSrc1Step,
const *Npp32s* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

One-channel 32-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus **nppiAverageRelativeError_32sc_C1R_Ctx**(const *Npp32sc* *pSrc1, int nSrc1Step,
const *Npp32sc* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError,
Npp8u *pDeviceBuffer,
NppStreamContext nppStreamCtx)

One-channel 32-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_32f_C1R_Ctx(const Npp32f *pSrc1, int nSrc1Step,
                                              const Npp32f *pSrc2, int nSrc2Step,
                                              NppiSize oSizeROI, Npp64f *pError, Npp8u
                                              *pDeviceBuffer, NppStreamContext
                                              nppStreamCtx)
```

One-channel 32-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_32fc_C1R_Ctx(const Npp32fc *pSrc1, int nSrc1Step,
                                              const Npp32fc *pSrc2, int nSrc2Step,
                                              NppiSize oSizeROI, Npp64f *pError,
                                              Npp8u *pDeviceBuffer,
                                              NppStreamContext nppStreamCtx)
```

One-channel 32-bit floating point complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_64f_C1R_Ctx(const Npp64f *pSrc1, int nSrc1Step,
                                              const Npp64f *pSrc2, int nSrc2Step,
                                              NppiSize oSizeROI, Npp64f *pError, Npp8u
                                              *pDeviceBuffer, NppStreamContext
                                              nppStreamCtx)
```

One-channel 64-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_8u_C2R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const
                                              Npp8u *pSrc2, int nSrc2Step, NppiSize
                                              oSizeROI, Npp64f *pError, Npp8u
                                              *pDeviceBuffer, NppStreamContext
                                              nppStreamCtx)
```

Two-channel 8-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_8s_C2R_Ctx(const Npp8s *pSrc1, int nSrc1Step, const
                                              Npp8s *pSrc2, int nSrc2Step, NppiSize
                                              oSizeROI, Npp64f *pError, Npp8u
                                              *pDeviceBuffer, NppStreamContext
                                              nppStreamCtx)
```

Two-channel 8-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_16u_C2R_Ctx(const Npp16u *pSrc1, int nSrc1Step,
                                              const Npp16u *pSrc2, int nSrc2Step,
                                              NppiSize oSizeROI, Npp64f *pError, Npp8u
                                              *pDeviceBuffer, NppStreamContext
                                              nppStreamCtx)
```

Two-channel 16-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_16s_C2R_Ctx(const Npp16s *pSrc1, int nSrc1Step,  
                                                const Npp16s *pSrc2, int nSrc2Step,  
                                                NppiSize oSizeROI, Npp64f *pError, Npp8u  
                                                *pDeviceBuffer, NppStreamContext  
                                                nppStreamCtx)
```

Two-channel 16-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_16sc_C2R_Ctx(const Npp16sc *pSrc1, int nSrc1Step,  
                                                const Npp16sc *pSrc2, int nSrc2Step,  
                                                NppiSize oSizeROI, Npp64f *pError,  
                                                Npp8u *pDeviceBuffer,  
                                                NppStreamContext nppStreamCtx)
```

Two-channel 16-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_32u_C2R_Ctx(const Npp32u *pSrc1, int nSrc1Step,  
                                                const Npp32u *pSrc2, int nSrc2Step,  
                                                NppiSize oSizeROI, Npp64f *pError, Npp8u  
                                                *pDeviceBuffer, NppStreamContext  
                                                nppStreamCtx)
```

Two-channel 32-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_32s_C2R_Ctx(const Npp32s *pSrc1, int nSrc1Step,  
                                                const Npp32s *pSrc2, int nSrc2Step,  
                                                NppiSize oSizeROI, Npp64f *pError, Npp8u  
                                                *pDeviceBuffer, NppStreamContext  
                                                nppStreamCtx)
```

Two-channel 32-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_32sc_C2R_Ctx(const Npp32sc *pSrc1, int nSrc1Step,  
                                                const Npp32sc *pSrc2, int nSrc2Step,  
                                                NppiSize oSizeROI, Npp64f *pError,  
                                                Npp8u *pDeviceBuffer,  
                                                NppStreamContext nppStreamCtx)
```

Two-channel 32-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_32f_C2R_Ctx(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 32-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_32fc_C2R_Ctx(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 32-bit floating point complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_64f_C2R_Ctx(const *Npp64f* *pSrc1, int nSrc1Step, const *Npp64f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Two-channel 64-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_8u_C3R_Ctx(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_8s_C3R_Ctx(const *Npp8s* *pSrc1, int nSrc1Step, const *Npp8s* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_16u_C3R_Ctx(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 16-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_16s_C3R_Ctx(const Npp16s *pSrc1, int nSrc1Step,  
const Npp16s *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError, Npp8u  
*pDeviceBuffer, NppStreamContext  
nppStreamCtx)
```

Three-channel 16-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_16sc_C3R_Ctx(const Npp16sc *pSrc1, int nSrc1Step,  
const Npp16sc *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError,  
Npp8u *pDeviceBuffer,  
NppStreamContext nppStreamCtx)
```

Three-channel 16-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_32u_C3R_Ctx(const Npp32u *pSrc1, int nSrc1Step,  
const Npp32u *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError, Npp8u  
*pDeviceBuffer, NppStreamContext  
nppStreamCtx)
```

Three-channel 32-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_32s_C3R_Ctx(const Npp32s *pSrc1, int nSrc1Step,  
const Npp32s *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError, Npp8u  
*pDeviceBuffer, NppStreamContext  
nppStreamCtx)
```

Three-channel 32-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_32sc_C3R_Ctx(const Npp32sc *pSrc1, int nSrc1Step,  
const Npp32sc *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError,  
Npp8u *pDeviceBuffer,  
NppStreamContext nppStreamCtx)
```

Three-channel 32-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_32f_C3R_Ctx(const Npp32f *pSrc1, int nSrc1Step,  
const Npp32f *pSrc2, int nSrc2Step,  
NppiSize oSizeROI, Npp64f *pError, Npp8u  
*pDeviceBuffer, NppStreamContext  
nppStreamCtx)
```

Three-channel 32-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_32fc_C3R_Ctx(const Npp32fc *pSrc1, int nSrc1Step,
                                                  const Npp32fc *pSrc2, int nSrc2Step,
                                                  NppiSize oSizeROI, Npp64f *pError,
                                                  Npp8u *pDeviceBuffer,
                                                  NppStreamContext nppStreamCtx)
```

Three-channel 32-bit floating point complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_64f_C3R_Ctx(const Npp64f *pSrc1, int nSrc1Step,
                                                  const Npp64f *pSrc2, int nSrc2Step,
                                                  NppiSize oSizeROI, Npp64f *pError, Npp8u
                                                  *pDeviceBuffer, NppStreamContext
                                                  nppStreamCtx)
```

Three-channel 64-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_8u_C4R_Ctx(const Npp8u *pSrc1, int nSrc1Step, const
                                                  Npp8u *pSrc2, int nSrc2Step, NppiSize
                                                  oSizeROI, Npp64f *pError, Npp8u
                                                  *pDeviceBuffer, NppStreamContext
                                                  nppStreamCtx)
```

Four-channel 8-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_8s_C4R_Ctx(const Npp8s *pSrc1, int nSrc1Step, const
                                                  Npp8s *pSrc2, int nSrc2Step, NppiSize
                                                  oSizeROI, Npp64f *pError, Npp8u
                                                  *pDeviceBuffer, NppStreamContext
                                                  nppStreamCtx)
```

Four-channel 8-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

```
NppStatus nppiAverageRelativeError_16u_C4R_Ctx(const Npp16u *pSrc1, int nSrc1Step,
                                                  const Npp16u *pSrc2, int nSrc2Step,
                                                  NppiSize oSizeROI, Npp64f *pError, Npp8u
                                                  *pDeviceBuffer, NppStreamContext
                                                  nppStreamCtx)
```

Four-channel 16-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_16s_C4R_Ctx(const *Npp16s* *pSrc1, int nSrc1Step,
const *Npp16s* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

Four-channel 16-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_16sc_C4R_Ctx(const *Npp16sc* *pSrc1, int nSrc1Step,
const *Npp16sc* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError,
Npp8u *pDeviceBuffer,
NppStreamContext nppStreamCtx)

Four-channel 16-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_32u_C4R_Ctx(const *Npp32u* *pSrc1, int nSrc1Step,
const *Npp32u* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

Four-channel 32-bit unsigned image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_32s_C4R_Ctx(const *Npp32s* *pSrc1, int nSrc1Step,
const *Npp32s* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

Four-channel 32-bit signed image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_32sc_C4R_Ctx(const *Npp32sc* *pSrc1, int nSrc1Step,
const *Npp32sc* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError,
Npp8u *pDeviceBuffer,
NppStreamContext nppStreamCtx)

Four-channel 32-bit signed complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_32f_C4R_Ctx(const *Npp32f* *pSrc1, int nSrc1Step,
const *Npp32f* *pSrc2, int nSrc2Step,
NppiSize oSizeROI, *Npp64f* *pError, *Npp8u*
*pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

Four-channel 32-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_32fc_C4R_Ctx(const *Npp32fc* *pSrc1, int nSrc1Step, const *Npp32fc* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 32-bit floating point complex image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

NppStatus nppiAverageRelativeError_64f_C4R_Ctx(const *Npp64f* *pSrc1, int nSrc1Step, const *Npp64f* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp64f* *pError, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 64-bit floating point image MaximumRelative_Error.

For common parameter descriptions, see *Common parameters for nppiMaximumError functions include:*.

AverageRelativeErrorGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff_Inf primitives.

1.14.38.2 CommonAverageRelativeErrorGetBufferHostSizeParameters

Common parameters for nppiAverageRelativeErrorGetBufferHostSize functions include:

param oSizeROI

Region-Of-Interest (ROI).

param hpBufferSize

Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer.*

param nppStreamCtx

Application Managed Stream Context.

return

NPP_NULL_POINTER_ERROR if hpBufferSize is 0 (NULL), *ROI Related Error Codes.*

NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C1R_Ctx(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiAverageRelativeError_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters.*

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_8s_C1R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_8s_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_16u_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_16u_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_16s_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_16s_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_16sc_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_16sc_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32u_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32u_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32s_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32s_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32sc_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32sc_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32f_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32f_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32fc_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32fc_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_64f_C1R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_64f_C1R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_8u_C2R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_8u_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_8s_C2R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_8s_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

```
NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C2R_Ctx(NppiSize oSizeROI,  
                                                                    size_t  
                                                                    *hpBufferSize,  
                                                                    NppStreamContext  
                                                                    nppStreamCtx)
```

Buffer size for *nppiAverageRelativeError_16u_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

```
NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C2R_Ctx(NppiSize oSizeROI,  
                                                                    size_t  
                                                                    *hpBufferSize,  
                                                                    NppStreamContext  
                                                                    nppStreamCtx)
```

Buffer size for *nppiAverageRelativeError_16s_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

```
NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C2R_Ctx(NppiSize oSizeROI,  
                                                                    size_t  
                                                                    *hpBufferSize,  
                                                                    NppStreamContext  
                                                                    nppStreamCtx)
```

Buffer size for *nppiAverageRelativeError_16sc_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

```
NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C2R_Ctx(NppiSize oSizeROI,  
                                                                    size_t  
                                                                    *hpBufferSize,  
                                                                    NppStreamContext  
                                                                    nppStreamCtx)
```

Buffer size for *nppiAverageRelativeError_32u_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

```
NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C2R_Ctx(NppiSize oSizeROI,  
                                                                    size_t  
                                                                    *hpBufferSize,  
                                                                    NppStreamContext  
                                                                    nppStreamCtx)
```

Buffer size for *nppiAverageRelativeError_32s_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus nppiAverageRelativeErrorGetBufferHostSize_32sc_C2R_Ctx(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32sc_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus nppiAverageRelativeErrorGetBufferHostSize_32f_C2R_Ctx(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32f_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R_Ctx(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32fc_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus nppiAverageRelativeErrorGetBufferHostSize_64f_C2R_Ctx(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_64f_C2R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus nppiAverageRelativeErrorGetBufferHostSize_8u_C3R_Ctx(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus nppiAverageRelativeErrorGetBufferHostSize_8s_C3R_Ctx(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_8s_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_16u_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_16u_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_16s_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_16s_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_16sc_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32u_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32u_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32s_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32s_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32sc_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32f_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32f_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32fc_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_64f_C3R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_64f_C3R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_8s_C4R_Ctx**(*NppiSize* oSizeROI,
size_t *hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_8s_C4R_Ctx*.

For common parameter descriptions, see [CommonAverageRelativeErrorGetBufferHostSizeParameters](#).

```
NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C4R_Ctx(NppiSize oSizeROI,  
                                                                size_t  
                                                                *hpBufferSize,  
                                                                NppStreamContext  
                                                                nppStreamCtx)
```

Buffer size for *nppiAverageRelativeError_16u_C4R_Ctx*.

For common parameter descriptions, see [CommonAverageRelativeErrorGetBufferHostSizeParameters](#).

```
NppStatus nppiAverageRelativeErrorGetBufferHostSize_16s_C4R_Ctx(NppiSize oSizeROI,  
                                                                size_t  
                                                                *hpBufferSize,  
                                                                NppStreamContext  
                                                                nppStreamCtx)
```

Buffer size for *nppiAverageRelativeError_16s_C4R_Ctx*.

For common parameter descriptions, see [CommonAverageRelativeErrorGetBufferHostSizeParameters](#).

```
NppStatus nppiAverageRelativeErrorGetBufferHostSize_16sc_C4R_Ctx(NppiSize oSizeROI,  
                                                                size_t  
                                                                *hpBufferSize,  
                                                                NppStreamContext  
                                                                nppStreamCtx)
```

Buffer size for *nppiAverageRelativeError_16sc_C4R_Ctx*.

For common parameter descriptions, see [CommonAverageRelativeErrorGetBufferHostSizeParameters](#).

```
NppStatus nppiAverageRelativeErrorGetBufferHostSize_32u_C4R_Ctx(NppiSize oSizeROI,  
                                                                size_t  
                                                                *hpBufferSize,  
                                                                NppStreamContext  
                                                                nppStreamCtx)
```

Buffer size for *nppiAverageRelativeError_32u_C4R_Ctx*.

For common parameter descriptions, see [CommonAverageRelativeErrorGetBufferHostSizeParameters](#).

```
NppStatus nppiAverageRelativeErrorGetBufferHostSize_32s_C4R_Ctx(NppiSize oSizeROI,  
                                                                size_t  
                                                                *hpBufferSize,  
                                                                NppStreamContext  
                                                                nppStreamCtx)
```

Buffer size for *nppiAverageRelativeError_32s_C4R_Ctx*.

For common parameter descriptions, see [CommonAverageRelativeErrorGetBufferHostSizeParameters](#).

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32sc_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32sc_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32f_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32f_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_32fc_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_32fc_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

NppStatus **nppiAverageRelativeErrorGetBufferHostSize_64f_C4R_Ctx**(*NppiSize* oSizeROI,
size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for *nppiAverageRelativeError_64f_C4R_Ctx*.

For common parameter descriptions, see *CommonAverageRelativeErrorGetBufferHostSizeParameters*.

1.14.39. Image Quality Assessment IQA

1.14.39.1 IQA

Primitives for computing the image quality between two images, such as MSE, PSNR, SSIM, and MS-SSIM.

MSE

NppStatus **nppiMSE_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp32f* *pMSE, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image MSE.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pMSE** – Device memory pointer to the computed MSE of two images.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiMSE_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp32f* *pMSE, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image MSE.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pMSE** – Device memory pointer to the computed MSE of two images.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

PSNR

NppStatus **nppiPSNR_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp32f* *pPSNR, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image PSNR_Ctx.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pPSNR** – Device memory pointer to the computed PSNR of two images.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiPSNR_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp32f* *pPSNR, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image PSNR_Ctx.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pPSNR** – Device memory pointer to the computed PSNR of two images.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

SSIM

NppStatus **nppiSSIM_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp32f* *pSSIM, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image SSIM.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pSSIM** – Device memory pointer to the computed SSIM of two images.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiSSIM_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp32f* *pSSIM, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image SSIM.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pSSIM** – Device memory pointer to the computed SSIM of two images.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

MSSSIM

NppStatus **nppiMSSSIM_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp32f* *pMSSSIM, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image MS-SSIM*.

This function will be deprecated in a future release use the nppiWMSSSIM functions instead.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pMSSSIM** – Device memory pointer to the computed MS-SSIM of two images.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

WMSSSIM

NppStatus **nppiWMSSSIM_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp32f* *pMSSSIM, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

One-channel 8-bit unsigned image MS-SSIM*.

This function uses the algorithm described in the paper by Wang et. al. Wang, Z., Simoncelli, E.P., Bovik, A.C. Multiscale Structural Similarity for Image Quality Assessment. In: The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, 13981402. Pacific Grove, CA, USA: IEEE,

- a. <https://doi.org/10.1109/ACSSC.2003.1292216>. NOTE: this API call can only process oSizeROI dimensions 16px by 16px and above. Any oSizeROI dimensions less than 16px by 16px will result in undefined behaviour.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pMSSSIM** – Device memory pointer to the computed MS-SSIM of two images.

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiWMSSSIM_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp32f* *pMSSSIM, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Three-channel 8-bit unsigned image MS-SSIM*.

This function uses the algorithm described in the paper by Wang et. al. Wang, Z., Simoncelli, E.P., Bovik, A.C. Multiscale Structural Similarity for Image Quality Assessment. In: The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, 13981402. Pacific Grove, CA, USA: IEEE,

- <https://doi.org/10.1109/ACSSC.2003.1292216>. NOTE: this API call can only process oSizeROI dimensions 16px by 16px and above. Any oSizeROI dimensions less than 16px by 16px will result in undefined behaviour.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer*.
- ▶ **nSrc1Step** – *Source-Image Line Step*.
- ▶ **pSrc2** – *Source-Image Pointer*.
- ▶ **nSrc2Step** – *Source-Image Line Step*.
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)*.
- ▶ **pMSSSIM** – Device memory pointer to the computed MS-SSIM of two images.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes.

NppStatus **nppiWMSSSIM_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, *Npp32f* *pMSSSIM, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Four-channel 8-bit unsigned image MS-SSIM*.

This function uses the algorithm described in the paper by Wang et. al. Wang, Z., Simoncelli, E.P., Bovik, A.C. Multiscale Structural Similarity for Image Quality Assessment. In: The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, 13981402. Pacific Grove, CA, USA: IEEE,

- <https://doi.org/10.1109/ACSSC.2003.1292216>. NOTE: this API call can only process oSizeROI dimensions 16px by 16px and above. Any oSizeROI dimensions less than 16px by 16px will result in undefined behaviour.

Parameters

- ▶ **pSrc1** – *Source-Image Pointer.*
- ▶ **nSrc1Step** – *Source-Image Line Step.*
- ▶ **pSrc2** – *Source-Image Pointer.*
- ▶ **nSrc2Step** – *Source-Image Line Step.*
- ▶ **oSizeROI** – *Region-Of-Interest (ROI).*
- ▶ **pMSSSIM** – Device memory pointer to the computed MS-SSIM of two images.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Image Data Related Error Codes, ROI Related Error Codes.

MSEGetBufferHostSize

NppStatus **nppiMSEGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMSE_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiMSEGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMSE_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiPSNRGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiPSNR_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiPSNRGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiPSNR_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiSSIMGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSSIM_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiSSIMGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiSSIM_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiMSSSIMGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiMSSSIM_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiWMSSSIMGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiWMSSSIM_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiWMSSSIMGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiWMSSSIM_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiWMSSSIMGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for *nppiWMSSSIM_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

1.14.40. Image Batch Quality Assessment

1.14.40.1 IQABatch

Primitives for computing the image quality for a batch of image pairs, such as MSE, PSNR, SSIM, and MS-SSIM with a single *Region-Of-Interest (ROI)* for all pairs of input images

MSEBatch

NppStatus **nppiMSEBatch_8u_C1R_Ctx**(const *NppImageDescriptor* *pSrc1BatchList, const *NppImageDescriptor* *pSrc2BatchList, int nBatchSize, *NppiSize* oSizeROI, *Npp32f* *pMSE, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned MSE for a batch of image pairs for a single ROI.

Provided oSizeROI will be used for all images passed in pSrc1BatchList and pSrc2BatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.

- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* ROI width and height of ALL images in the batch, MUST match the ROI used when the label markers UF image was generated.
- ▶ **pMSE** – Device memory pointer to output array of the computed MSE for nBatchSize * sizeof(Npp32f) * 1 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiMSEBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus nppiMSEBatch_8u_C3R_Ctx(const *NppImageDescriptor* *pSrc1BatchList, const *NppImageDescriptor* *pSrc2BatchList, int nBatchSize, *NppiSize* oSizeROI, *Npp32f* *pMSE, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned MSE for a batch of image pairs for a single ROI.

Provided oSizeROI will be used for all images passed in pSrc1BatchList and pSrc2BatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* ROI width and height of ALL images in the batch, MUST match the ROI used when the label markers UF image was generated.
- ▶ **pMSE** – Device memory pointer to output array of the computed MSE for nBatchSize * sizeof(Npp32f) * 3 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiMSEBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

PSNRBatch

NppStatus **nppiPSNRBatch_8u_C1R_Ctx**(const *NppImageDescriptor* *pSrc1BatchList, const *NppImageDescriptor* *pSrc2BatchList, int nBatchSize, *NppiSize* oSizeROI, *Npp32f* *pPSNR, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned PSNR for a batch of image pairs for a single ROI.

Provided oSizeROI will be used for all images passed in pSrc1BatchList and pSrc2BatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* ROI width and height of ALL images in the batch, MUST match the ROI used when the label markers UF image was generated.
- ▶ **pPSNR** – Device memory pointer to output array of the computed PSNR for nBatchSize * sizeof(Npp32f) * 1 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiPSNRBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiPSNRBatch_8u_C3R_Ctx**(const *NppImageDescriptor* *pSrc1BatchList, const *NppImageDescriptor* *pSrc2BatchList, int nBatchSize, *NppiSize* oSizeROI, *Npp32f* *pPSNR, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned PSNR for a batch of image pairs for a single ROI.

Provided oSizeROI will be used for all images passed in pSrc1BatchList and pSrc2BatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).

- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* ROI width and height of ALL images in the batch, MUST match the ROI used when the label markers UF image was generated.
- ▶ **pPSNR** – Device memory pointer to output array of the computed PSNR for $nBatchSize * sizeof(Npp32f) * 3$ image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one *nppiPSNRBatchGetBufferHostSize* call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

SSIMBatch

NppStatus **nppiSSIMBatch_8u_C1R_Ctx**(const *NppImageDescriptor* *pSrc1BatchList, const *NppImageDescriptor* *pSrc2BatchList, int nBatchSize, *NppiSize* oSizeROI, *Npp32f* *pSSIM, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned SSIM for a batch of image pairs for a single ROI.

Provided oSizeROI will be used for all images passed in pSrc1BatchList and pSrc2BatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* ROI width and height of ALL images in the batch, MUST match the ROI used when the label markers UF image was generated.
- ▶ **pSSIM** – Device memory pointer to output array of the computed SSIM for $nBatchSize * sizeof(Npp32f) * 1$ image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one *nppiSSIMBatchGetBufferHostSize* call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiSSIMBatch_8u_C3R_Ctx**(const *NppImageDescriptor* *pSrc1BatchList, const *NppImageDescriptor* *pSrc2BatchList, int nBatchSize, *NppiSize* oSizeROI, *Npp32f* *pSSIM, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned SSIM for a batch of image pairs for a single ROI.

Provided oSizeROI will be used for all images passed in pSrc1BatchList and pSrc2BatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* ROI width and height of ALL images in the batch, MUST match the ROI used when the label markers UF image was generated.
- ▶ **pSSIM** – Device memory pointer to output array of the computed SSIM for nBatchSize * sizeof(Npp32f) * 3 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiSSIMBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

WMSSSIMBatch

NppStatus nppiWMSSSIMBatch_8u_C1R_Ctx(const *NppImageDescriptor* *pSrc1BatchList, const *NppImageDescriptor* *pSrc2BatchList, int nBatchSize, *NppiSize* oSizeROI, *Npp32f* *pWMSSSIM, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned WMSSSIM for a batch of image pairs for a single ROI.

Provided oSizeROI will be used for all images passed in pSrc1BatchList and pSrc2BatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images. NOTE: this API call can only process oSizeROI dimensions 16 pixels by 16 pixels and above. Any oSizeROI dimensions less than 16 pixels by 16 pixels will result in undefined behaviour.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.

- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* ROI width and height of ALL images in the batch, MUST match the ROI used when the label markers UF image was generated.
- ▶ **pWMSSSIM** – Device memory pointer to output array of the computed WMSSSIM for nBatchSize * sizeof(Npp32f) * 1 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiWMSSSIMBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus nppiWMSSSIMBatch_8u_C3R_Ctx(const *NppImageDescriptor* *pSrc1BatchList, const *NppImageDescriptor* *pSrc2BatchList, int nBatchSize, *NppiSize* oSizeROI, *Npp32f* *pWMSSSIM, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned WMSSSIM for a batch of image pairs for a single ROI.

Provided oSizeROI will be used for all images passed in pSrc1BatchList and pSrc2BatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images. NOTE: this API call can only process ROI dimensions 16 pixels by 16 pixels and above. Any ROI dimensions less than 16 pixels by 16 pixels will result in undefined behaviour.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* ROI width and height of ALL images in the batch, MUST match the ROI used when the label markers UF image was generated.
- ▶ **pWMSSSIM** – Device memory pointer to output array of the computed WMSSSIM for nBatchSize * sizeof(Npp32f) * 3 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiWMSSSIMBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus **nppiWMSSSIMBatch_8u_C4R_Ctx**(const *NppImageDescriptor* *pSrc1BatchList, const *NppImageDescriptor* *pSrc2BatchList, int nBatchSize, *NppiSize* oSizeROI, *Npp32f* *pWMSSSIM, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned WMSSSIM for a batch of image pairs for a single ROI.

Provided oSizeROI will be used for all images passed in pSrc1BatchList and pSrc2BatchList arguments. API user must ensure that provided ROI (oSizeROI) does not go beyond the borders of any of provided images. NOTE: this API call can only process ROI dimensions 16 pixels by 16 pixels and above. Any ROI dimensions less than 16 pixels by 16 pixels will result in undefined behaviour.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors, per image oSize must be initialized.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oSizeROI** – *Region-Of-Interest (ROI)* ROI width and height of ALL images in the batch, MUST match the ROI used when the label markers UF image was generated.
- ▶ **pWMSSSIM** – Device memory pointer to output array of the computed WMSSSIM for nBatchSize * sizeof(Npp32f) * 4 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiWMSSSIMBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

MSEBatchGetBufferHostSize

NppStatus **nppiMSEBatchGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for a single image pair in the batch of image pairs for *nppiMSEBatch_8u_C1R_Ctx* For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiMSEBatchGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for a single image pair in the batch of image pairs for *nppiMSEBatch_8u_C3R_Ctx* For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiPSNRBatchGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Buffer size for a single image pair in the batch of image pairs for *nppiPSNRBatch_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiPSNRBatchGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for a single image pair in the batch of image pairs for *nppiPSNRBatch_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiSSIMBatchGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for a single image pair in the batch of image pairs for *nppiSSIMBatch_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiSSIMBatchGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize, *NppStreamContext*
nppStreamCtx)

Buffer size for a single image pair in the batch of image pairs for *nppiSSIMBatch_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiWMSSSIMBatchGetBufferHostSize_8u_C1R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for a single image pair in the batch of image pairs for *nppiWMSSSIM-Batch_8u_C1R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiWMSSSIMBatchGetBufferHostSize_8u_C3R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for a single image pair in the batch of image pairs for *nppiWMSSSIM-Batch_8u_C3R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

NppStatus **nppiWMSSSIMBatchGetBufferHostSize_8u_C4R_Ctx**(*NppiSize* oSizeROI, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Buffer size for a single image pair in the batch of image pairs for *nppiWMSSSIM-Batch_8u_C4R_Ctx*.

For common parameter descriptions, see *CommonGetBufferHostSizeParameters*.

1.14.41. Image Advanced Batch Quality Assessment

1.14.41.1 IQABatchAdvanced

Primitives for computing the image quality for a batch of image pairs, such as MSE, PSNR, SSIM, and MS-SSIM with per-image *Region-Of-Interest (ROI)*

MSEBatchAdvanced

```
NppStatus nppiMSEBatch_8u_C1R_Advanced_Ctx( const NppImageDescriptor *pSrc1BatchList,
                                              const NppImageDescriptor *pSrc2BatchList,
                                              int nBatchSize, NppiSize oMaxSizeROI, Npp32f
                                              *pMSE, NppiBufferDescriptor
                                              *pDeviceBufferList, NppStreamContext
                                              nppStreamCtx )
```

1 channel 8-bit unsigned MSE for a batch of image pairs with per-image ROI

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)* maximum ROI width and height of ALL images in the batch.
- ▶ **pMSE** – Device memory pointer to output array of the computed MSE for nBatchSize * sizeof(Npp32f * 1 image pairs).
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiMSEBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiMSEBatch_8u_C3R_Advanced_Ctx( const NppImageDescriptor *pSrc1BatchList,
                                              const NppImageDescriptor *pSrc2BatchList,
                                              int nBatchSize, NppiSize oMaxSizeROI, Npp32f
                                              *pMSE, NppiBufferDescriptor
                                              *pDeviceBufferList, NppStreamContext
                                              nppStreamCtx )
```

3 channel 8-bit unsigned MSE for a batch of image pairs with per-image ROI

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.

- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)* maximum ROI width and height of ALL images in the batch.
- ▶ **pMSE** – Device memory pointer to output array of the computed MSE for $nBatchSize * \text{sizeof}(Npp32f) * 3$ image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one `nppiMSEBatchGetBufferHostSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

PSNRBatchAdvanced

NppStatus `nppiPSNRBatch_8u_C1R_Advanced_Ctx`(const *NppImageDescriptor* *pSrc1BatchList,
const *NppImageDescriptor* *pSrc2BatchList,
int nBatchSize, *NppiSize* oMaxSizeROI,
Npp32f *pPSNR, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned PSNR for a batch of image pairs with per-image ROI

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)* maximum ROI width and height of ALL images in the batch.
- ▶ **pPSNR** – Device memory pointer to output array of the computed PSNR for $nBatchSize * \text{sizeof}(Npp32f) * 1$ image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one `nppiPSNRBatchGetBufferHostSize` call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes


```
NppStatus nppiPSNRBatch_8u_C3R_Advanced_Ctx(const NppImageDescriptor *pSrc1BatchList,
                                             const NppImageDescriptor *pSrc2BatchList,
                                             int nBatchSize, NppiSize oMaxSizeROI,
                                             Npp32f *pPSNR, NppiBufferDescriptor
                                             *pDeviceBufferList, NppStreamContext
                                             nppStreamCtx)
```

3 channel 8-bit unsigned PSNR for a batch of image pairs with per-image ROI

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)* maximum ROI width and height of ALL images in the batch.
- ▶ **pPSNR** – Device memory pointer to output array of the computed PSNR for nBatchSize * sizeof(Npp32f) * 3 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiPSNRBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

WMSSSIMBatchAdvanced

```
NppStatus nppiWMSSSIMBatch_8u_C1R_Advanced_Ctx(const NppImageDescriptor
                                                *pSrc1BatchList, const
                                                NppImageDescriptor *pSrc2BatchList, int
                                                nBatchSize, NppiSize oMaxSizeROI,
                                                Npp32f *pWMSSSIM,
                                                NppiBufferDescriptor *pDeviceBufferList,
                                                NppStreamContext nppStreamCtx)
```

1 channel 8-bit unsigned WMSSSIM for a batch of image pairs with per-image ROI NOTE: It is the user's responsibility to make sure the dimensions of per-image ROIs are 16 pixels by 16 pixels and above.

Any per-image ROI dimensions less than 16 pixels by 16 pixels will result in undefined behaviour.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.

- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)* maximum ROI width and height of ALL images in the batch.
- ▶ **pWMSSSIM** – Device memory pointer to output array of the computed WMSSSIM for nBatchSize * sizeof(Npp32f) * 1 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiWMSSSIMBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

NppStatus nppiWMSSSIMBatch_8u_C3R_Advanced_Ctx(const *NppImageDescriptor* *pSrc1BatchList, const *NppImageDescriptor* *pSrc2BatchList, int nBatchSize, *NppiSize* oMaxSizeROI, *Npp32f* *pWMSSSIM, *NppiBufferDescriptor* *pDeviceBufferList, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned WMSSSIM for a batch of image pairs with per-image ROI NOTE: It is the user's responsibility to make sure the dimensions of per-image ROIs are 16 pixels by 16 pixels and above.

Any per-image ROI dimensions less than 16 pixels by 16 pixels will result in undefined behaviour.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)* maximum ROI width and height of ALL images in the batch.
- ▶ **pWMSSSIM** – Device memory pointer to output array of the computed WMSSSIM for nBatchSize * sizeof(Npp32f) * 3 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiWMSSSIMBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

```
NppStatus nppiWMSSSIMBatch_8u_C4R_Advanced_Ctx(const NppImageDescriptor
                                                *pSrc1BatchList, const
                                                NppImageDescriptor *pSrc2BatchList, int
                                                nBatchSize, NppiSize oMaxSizeROI,
                                                Npp32f *pWMSSSIM,
                                                NppiBufferDescriptor *pDeviceBufferList,
                                                NppStreamContext nppStreamCtx)
```

4 channel 8-bit unsigned WMSSSIM for a batch of image pairs with per-image ROI NOTE: It is the user's responsibility to make sure the dimensions of per-image ROIs are 16 pixels by 16 pixels and above.

Any per-image ROI dimensions less than 16 pixels by 16 pixels will result in undefined behaviour.

Parameters

- ▶ **pSrc1BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **pSrc2BatchList** – *Source-Batch-Images Pointer* device memory pointer to the list of device memory image descriptors.
- ▶ **nBatchSize** – Number of *NppImageDescriptor*, *NppiBufferDescriptor*, and new max number structures/values processed in this call (must be > 1).
- ▶ **oMaxSizeROI** – *Region-Of-Interest (ROI)* maximum ROI width and height of ALL images in the batch.
- ▶ **pWMSSSIM** – Device memory pointer to output array of the computed WMSSSIM for nBatchSize * sizeof(Npp32f) * 4 image pairs.
- ▶ **pDeviceBufferList** – Device memory pointer to the list of *NppiBufferDescriptor* buffer descriptors specifying per image device memory buffer pointers and size as returned by at least one nppiWMSSSIMBatchGetBufferHostSize call.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Image Data Related Error Codes, ROI Related Error Codes

1.15. Image Threshold And Compare Operations

Methods for pixel-wise threshold and compare operations.

These functions can be found in the nppitc library. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

1.15.1. Image Threshold Operations

1.15.1.1 Threshold Operations

Threshold image pixels.

1.15.1.1.1 Common parameters for nppiThreshold non-inplace and inplace functions:

param pSrcDst
In-Place Image Pointer for inplace functions.

param nSrcDstStep
In-Place-Image Line Step for inplace functions.

param pSrc
Source-Image Pointer for non-inplace functions.

param nSrcStep
Source-Image Line Step for non-inplace functions.

param pDst
Destination-Image Pointer for non-inplace functions.

param nDstStep
Destination-Image Line Step for non-inplace functions.

param oSizeROI
Region-Of-Interest (ROI).

param nThreshold
 The threshold value.

param eComparisonOperation
 The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes, or NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

Functions

NppStatus **nppiThreshold_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions*.

NppStatus nppiThreshold_8u_C1IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions*.

NppStatus nppiThreshold_16u_C1IR_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions*.

NppStatus nppiThreshold_16u_C1IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions*.

NppStatus nppiThreshold_16s_C1IR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* nThreshold, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions*.

NppStatus nppiThreshold_16s_C1IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* nThreshold, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* nThreshold, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_32f_C1IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* nThreshold, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_8u_C3IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_16u_C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_16u_C3IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_16s_C3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_16s_C3IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_32f_C3IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_8u_AC4IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_16u_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_16u_AC4IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions:*.

NppStatus nppiThreshold_16s_AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions*.

NppStatus **nppiThreshold_16s_AC4IR_Ctx**(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions*.

NppStatus **nppiThreshold_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions*.

NppStatus **nppiThreshold_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold non-inplace and inplace functions*.

1.15.1.2 Image Threshold Greater Than Operations

1.15.1.2.1 Threshold Greater Than Operations

Threshold greater than image pixels.

1.15.1.2.1.1 Common parameters for nppiThreshold_GT non-inplace and inplace functions:

param pSrcDst
In-Place Image Pointer for inplace functions.

param nSrcDstStep
In-Place-Image Line Step for inplace functions.

param pSrc
Source-Image Pointer for non-inplace functions.

param nSrcStep

Source-Image Line Step for non-inplace functions.

param pDst

Destination-Image Pointer for non-inplace functions.

param nDstStep

Destination-Image Line Step for non-inplace functions.

param oSizeROI

Region-Of-Interest (ROI).

param nThreshold

The threshold value.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes.

Functions

NppStatus **nppiThreshold_GT_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*

NppStatus **nppiThreshold_GT_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*

NppStatus **nppiThreshold_GT_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*

NppStatus **nppiThreshold_GT_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_GT_16s_C1R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp16s  
nThreshold, NppStreamContext nppStreamCtx)
```

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_GT_16s_C1IR_Ctx(Npp16s *pSrcDst, int nSrcDstStep, NppiSize  
oSizeROI, const Npp16s nThreshold,  
NppStreamContext nppStreamCtx)
```

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_GT_32f_C1R_Ctx(const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp32f  
nThreshold, NppStreamContext nppStreamCtx)
```

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_GT_32f_C1IR_Ctx(Npp32f *pSrcDst, int nSrcDstStep, NppiSize  
oSizeROI, const Npp32f nThreshold,  
NppStreamContext nppStreamCtx)
```

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_GT_8u_C3R_Ctx(const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int  
nDstStep, NppiSize oSizeROI, const Npp8u  
rThresholds[3], NppStreamContext nppStreamCtx)
```

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

NppStatus nppiThreshold_GT_8u_C3IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

NppStatus nppiThreshold_GT_16u_C3IR_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

NppStatus nppiThreshold_GT_16u_C3IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

NppStatus nppiThreshold_GT_16s_C3IR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

NppStatus nppiThreshold_GT_16s_C3IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions:*.

NppStatus nppiThreshold_GT_32f_C3IR_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions*.

```
NppStatus nppiThreshold_GT_32f_C3IR_Ctx(Npp32f *pSrcDst, int nSrcDstStep, NppiSize
                                         oSizeROI, const Npp32f rThresholds[3],
                                         NppStreamContext nppStreamCtx)
```

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions*.

```
NppStatus nppiThreshold_GT_8u_AC4R_Ctx(const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int
                                         nDstStep, NppiSize oSizeROI, const Npp8u
                                         rThresholds[3], NppStreamContext nppStreamCtx)
```

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions*.

```
NppStatus nppiThreshold_GT_8u_AC4IR_Ctx(Npp8u *pSrcDst, int nSrcDstStep, NppiSize
                                         oSizeROI, const Npp8u rThresholds[3],
                                         NppStreamContext nppStreamCtx)
```

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions*.

```
NppStatus nppiThreshold_GT_16u_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u *pDst,
                                         int nDstStep, NppiSize oSizeROI, const Npp16u
                                         rThresholds[3], NppStreamContext
                                         nppStreamCtx)
```

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions*.

```
NppStatus nppiThreshold_GT_16u_AC4IR_Ctx(Npp16u *pSrcDst, int nSrcDstStep, NppiSize
                                         oSizeROI, const Npp16u rThresholds[3],
                                         NppStreamContext nppStreamCtx)
```

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions*.

NppStatus nppiThreshold_GT_16s_AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions*.

NppStatus nppiThreshold_GT_16s_AC4IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions*.

NppStatus nppiThreshold_GT_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions*.

NppStatus nppiThreshold_GT_32f_AC4IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GT non-inplace and inplace functions*.

1.15.1.3 Image Threshold Less Than Operations

1.15.1.3.1 Threshold Less Than Operations

Threshold less than image pixels.

1.15.1.3.1.1 Common parameters for `nppiThreshold_LT` non-inplace and inplace functions:

param pSrcDst
In-Place Image Pointer for inplace functions.

param nSrcDstStep
In-Place-Image Line Step for inplace functions.

param pSrc
Source-Image Pointer for non-inplace functions.

param nSrcStep
Source-Image Line Step for non-inplace functions.

param pDst
Destination-Image Pointer for non-inplace functions.

param nDstStep
Destination-Image Line Step for non-inplace functions.

param oSizeROI
Region-Of-Interest (ROI).

param nThreshold
The threshold value.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes.

Functions

NppStatus `nppiThreshold_LT_8u_C1R_Ctx`(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char threshold.

If for a comparison operations `sourcePixel` is less than `nThreshold` is true, the pixel is set to `nThreshold`, otherwise it is set to `sourcePixel`.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*

NppStatus `nppiThreshold_LT_8u_C1IR_Ctx`(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*.

NppStatus nppiThreshold_LT_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*.

NppStatus nppiThreshold_LT_16u_C1IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*.

NppStatus nppiThreshold_LT_16s_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*.

NppStatus nppiThreshold_LT_16s_C1IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*.

NppStatus nppiThreshold_LT_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_32f_C1IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* nThreshold, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_8u_C3IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_16u_C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_16u_C3IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_16s_C3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*.

NppStatus nppiThreshold_LT_16s_C3IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*.

NppStatus nppiThreshold_LT_32f_C3IR_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*.

NppStatus nppiThreshold_LT_32f_C3IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*.

NppStatus nppiThreshold_LT_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions:*.

NppStatus nppiThreshold_LT_8u_AC4IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_16u_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_16u_AC4IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_16s_AC4R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_16s_AC4IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus nppiThreshold_LT_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*.

NppStatus **nppiThreshold_LT_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LT non-inplace and inplace functions*:

1.15.1.4 Image Threshold Value Operations

1.15.1.4.1 Threshold Value Operations

Replace thresholded image pixels with a value.

1.15.1.4.1.1 Common parameters for nppiThreshold_Val non-inplace and inplace functions:

param pSrcDst

In-Place Image Pointer for inplace functions.

param nSrcDstStep

In-Place-Image Line Step for inplace functions.

param pSrc

Source-Image Pointer for non-inplace functions.

param nSrcStep

Source-Image Line Step for non-inplace functions.

param pDst

Destination-Image Pointer for non-inplace functions.

param nDstStep

Destination-Image Line Step for non-inplace functions.

param oSizeROI

Region-Of-Interest (ROI).

param nThreshold

The threshold value.

param nValue

The threshold replacement value.

param eComparisonOperation

The type of comparison operation to be used. The only valid values are: NPP_CMP_LESS and NPP_CMP_GREATER.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes, or

NPP_NOT_SUPPORTED_MODE_ERROR if an invalid comparison operation type is specified.

Functions

NppStatus **nppiThreshold_Val_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, const *Npp8u* nValue, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus **nppiThreshold_Val_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, const *Npp8u* nValue, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus **nppiThreshold_Val_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, const *Npp16u* nValue, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus **nppiThreshold_Val_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, const *Npp16u* nValue, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus nppiThreshold_Val_16s_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* nThreshold, const *Npp16s* nValue, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus nppiThreshold_Val_16s_C1IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* nThreshold, const *Npp16s* nValue, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus nppiThreshold_Val_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* nThreshold, const *Npp32f* nValue, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus nppiThreshold_Val_32f_C1IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* nThreshold, const *Npp32f* nValue, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus nppiThreshold_Val_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], const *Npp8u* rValues[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

```
NppStatus nppiThreshold_Val_8u_C3IR_Ctx(Npp8u *pSrcDst, int nSrcDstStep, NppiSize
                                         oSizeROI, const Npp8u rThresholds[3], const
                                         Npp8u rValues[3], NppCmpOp
                                         eComparisonOperation, NppStreamContext
                                         nppStreamCtx)
```

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

```
NppStatus nppiThreshold_Val_16u_C3R_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u *pDst,
                                         int nDstStep, NppiSize oSizeROI, const Npp16u
                                         rThresholds[3], const Npp16u rValues[3],
                                         NppCmpOp eComparisonOperation,
                                         NppStreamContext nppStreamCtx)
```

3 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

```
NppStatus nppiThreshold_Val_16u_C3IR_Ctx(Npp16u *pSrcDst, int nSrcDstStep, NppiSize
                                         oSizeROI, const Npp16u rThresholds[3], const
                                         Npp16u rValues[3], NppCmpOp
                                         eComparisonOperation, NppStreamContext
                                         nppStreamCtx)
```

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

```
NppStatus nppiThreshold_Val_16s_C3R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst,
                                         int nDstStep, NppiSize oSizeROI, const Npp16s
                                         rThresholds[3], const Npp16s rValues[3],
                                         NppCmpOp eComparisonOperation,
                                         NppStreamContext nppStreamCtx)
```

3 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus nppiThreshold_Val_16s_C3IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], const *Npp16s* rValues[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus nppiThreshold_Val_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus nppiThreshold_Val_32f_C3IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus nppiThreshold_Val_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], const *Npp8u* rValues[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*.

NppStatus nppiThreshold_Val_8u_AC4IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], const *Npp8u* rValues[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_Val_16u_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u *pDst,  
int nDstStep, NppiSize oSizeROI, const Npp16u  
rThresholds[3], const Npp16u rValues[3],  
NppCmpOp eComparisonOperation,  
NppStreamContext nppStreamCtx)
```

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_Val_16u_AC4IR_Ctx(Npp16u *pSrcDst, int nSrcDstStep, NppiSize  
oSizeROI, const Npp16u rThresholds[3], const  
Npp16u rValues[3], NppCmpOp  
eComparisonOperation, NppStreamContext  
nppStreamCtx)
```

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_Val_16s_AC4R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s *pDst,  
int nDstStep, NppiSize oSizeROI, const Npp16s  
rThresholds[3], const Npp16s rValues[3],  
NppCmpOp eComparisonOperation,  
NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_Val_16s_AC4IR_Ctx(Npp16s *pSrcDst, int nSrcDstStep, NppiSize  
oSizeROI, const Npp16s rThresholds[3], const  
Npp16s rValues[3], NppCmpOp  
eComparisonOperation, NppStreamContext  
nppStreamCtx)
```

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions:*.

NppStatus **nppiThreshold_Val_32f_AC4R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*:

NppStatus **nppiThreshold_Val_32f_AC4IR_Ctx**(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_Val non-inplace and inplace functions*:

1.15.1.5 Image Threshold Greater Than Value Operations

1.15.1.5.1 Threshold Greater Than Value Operations

Replace image pixels greater than threshold with a value.

1.15.1.5.1.1 Common parameters for nppiThreshold_GTVal non-inplace and inplace functions:

param pSrcDst

In-Place Image Pointer for inplace functions.

param nSrcDstStep

In-Place-Image Line Step for inplace functions.

param pSrc

Source-Image Pointer for non-inplace functions.

param nSrcStep

Source-Image Line Step for non-inplace functions.

param pDst

Destination-Image Pointer for non-inplace functions.

param nDstStep

Destination-Image Line Step for non-inplace functions.

param oSizeROI

Region-Of-Interest (ROI).

param nThreshold

The threshold value.

param nValue

The threshold replacement value.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes.

Functions

NppStatus **nppiThreshold_GTVa1_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, const *Npp8u* nValue, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus **nppiThreshold_GTVa1_8u_C1IR_Ctx**(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, const *Npp8u* nValue, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus **nppiThreshold_GTVa1_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, const *Npp16u* nValue, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus **nppiThreshold_GTVa1_16u_C1IR_Ctx**(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, const *Npp16u* nValue, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_GTVa1_16s_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* nThreshold, const *Npp16s* nValue, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_GTVa1_16s_C1IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* nThreshold, const *Npp16s* nValue, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_GTVa1_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* nThreshold, const *Npp32f* nValue, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_GTVa1_32f_C1IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* nThreshold, const *Npp32f* nValue, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_GTVa1_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], const *Npp8u* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_GTVa1_8u_C3IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], const *Npp8u* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_GTVa1_16u_C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, const *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], const *Npp16u* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_GTVa1_16u_C3IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], const *Npp16u* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_GTVa1_16s_C3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], const *Npp16s* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_GTVa1_16s_C3IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], const *Npp16s* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_GTVa1_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], const *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_GTVa1_32f_C3IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_GTVa1_8u_AC4R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], const *Npp8u* rValues[3], const *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_GTVa1_8u_AC4IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], const *Npp8u* rValues[3], *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_GTVa1_16u_AC4R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], const *Npp16u* rValues[3], const *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_GTVa1_16u_AC4IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], const *Npp16u* rValues[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_GTVa1_16s_AC4IR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], const *Npp16s* rValues[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_GTVa1_16s_AC4IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], const *Npp16s* rValues[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_GTVa1_32f_AC4IR_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_GTVa1_32f_AC4IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_GTVa1 non-inplace and inplace functions*:

1.15.1.6 Image Threshold Less Than Value Operations

1.15.1.6.1 Threshold Less Than Value Operations

Replace image pixels less than threshold with a value.

1.15.1.6.1.1 Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:

param pSrcDst
In-Place Image Pointer for inplace functions.

param nSrcDstStep
In-Place-Image Line Step for inplace functions.

param pSrc
Source-Image Pointer for non-inplace functions.

param nSrcStep
Source-Image Line Step for non-inplace functions.

param pDst
Destination-Image Pointer for non-inplace functions.

param nDstStep
Destination-Image Line Step for non-inplace functions.

param oSizeROI
Region-Of-Interest (ROI).

param nThreshold
 The threshold value.

param nValue
 The threshold replacement value.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes.

Functions

NppStatus **nppiThreshold_LTVa1_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, const *Npp8u* nValue, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_LTVa1_8u_C1IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* nThreshold, const *Npp8u* nValue, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_LTVa1_16u_C1IR_Ctx(const *Npp16u* *pSrc, int nSrcStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, const *Npp16u* nValue, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_LTVa1_16u_C1IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* nThreshold, const *Npp16u* nValue, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_LTVa1_16s_C1IR_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* nThreshold, const *Npp16s* nValue, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_LTVa1_16s_C1IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* nThreshold, const *Npp16s* nValue, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_LTVa1_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* nThreshold, const *Npp32f* nValue, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_LTVa1_32f_C1IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* nThreshold, const *Npp32f* nValue, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_LTVa1_8u_C3R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], const *Npp8u* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_LTVa1_8u_C3IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* rThresholds[3], const *Npp8u* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

NppStatus nppiThreshold_LTVa1_16u_C3R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], const *Npp16u* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1_16u_C3IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* rThresholds[3], const *Npp16u* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1_16s_C3R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], const *Npp16s* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1_16s_C3IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], const *Npp16s* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1_32f_C3R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1_32f_C3IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1_8u_AC4R_Ctx(const Npp8u *pSrc, int nSrcStep, Npp8u *pDst,
                                         int nDstStep, NppiSize oSizeROI, const Npp8u
                                         rThresholds[3], const Npp8u rValues[3],
                                         NppStreamContext nppStreamCtx)
```

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1_8u_AC4IR_Ctx(Npp8u *pSrcDst, int nSrcDstStep, NppiSize
                                         oSizeROI, const Npp8u rThresholds[3], const
                                         Npp8u rValues[3], NppStreamContext
                                         nppStreamCtx)
```

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1_16u_AC4R_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u
                                         *pDst, int nDstStep, NppiSize oSizeROI, const
                                         Npp16u rThresholds[3], const Npp16u
                                         rValues[3], NppStreamContext nppStreamCtx)
```

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1_16u_AC4IR_Ctx(Npp16u *pSrcDst, int nSrcDstStep, NppiSize
                                         oSizeROI, const Npp16u rThresholds[3], const
                                         Npp16u rValues[3], NppStreamContext
                                         nppStreamCtx)
```

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1_16s_AC4R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s
                                         *pDst, int nDstStep, NppiSize oSizeROI, const
                                         Npp16s rThresholds[3], const Npp16s
                                         rValues[3], NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1_16s_AC4IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* rThresholds[3], const *Npp16s* rValues[3], *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1_32f_AC4IR_Ctx(const *Npp32f* *pSrc, int nSrcStep, *NppiSize* oSizeROI, const *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1_32f_AC4IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* rThresholds[3], const *Npp32f* rValues[3], *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1 non-inplace and inplace functions*.

1.15.1.7 Image Fused AbsDiff Threshold Greater Than Value Operations

1.15.1.7.1 Fused AbsDiff Threshold Greater Than Value Operations

Replace image pixels greater than threshold with a value.

Supported data types include NPP_8U, NPP_16U, NPP_16S, NPP_32F. Supported channel counts include NPP_CH_1, NPP_CH_3, NPP_CH_A4.

param eSrcDstType
image data type.

param eSrcDstChannels
image channels.

param pSrcDst
In-Place Image Pointer for inplace functions.

param nSrcDstStep
In-Place-Image Line Step for inplace functions.

param pSrc1*Source-Image Pointer* for non-inplace functions.**param nSrc1Step***Source-Image Line Step* for non-inplace functions.**param pSrc2***Source-Image Pointer* to second source image for non-inplace functions.**param nSrc2Step***Source-Image Line Step* of second source image for non-inplace functions.**param pDst***Destination-Image Pointer* for non-inplace functions.**param nDstStep***Destination-Image Line Step* for non-inplace functions.**param oSizeROI***Region-Of-Interest (ROI).***param pThreshold**

The threshold value.

param pValue

The threshold replacement value.

param nppStreamCtx*Application Managed Stream Context.***return***Image Data Related Error Codes, ROI Related Error Codes.***Functions**

NppStatus **nppiFusedAbsDiff_Threshold_GTVal1_Ctx**(*NppDataType* eSrcDstType, *NppiChannels* eSrcDstChannels, const void *pSrc1, int nSrc1Step, const void *pSrc2, int nSrc2Step, void *pDst, int nDstStep, *NppiSize* oSizeROI, const void *pThreshold, const void *pvalue, *NppStreamContext* nppStreamCtx)

Image fused absdiff and greater than threshold value.

If for a comparison operations absdiff of sourcePixels is greater than pThreshold is true, the output pixel is set to pValue, otherwise it is set to absdiff of sourcePixels.

NppStatus **nppiFusedAbsDiff_Threshold_GTVal1_I_Ctx**(*NppDataType* eSrcDstType, *NppiChannels* eSrcDstChannels, void *pSrcDst, int nSrcDstStep, const void *pSrc2, int nSrc2Step, *NppiSize* oSizeROI, const void *pThreshold, const void *pvalue, *NppStreamContext* nppStreamCtx)

In place fused absdiff image greater than threshold value.

If for a comparison operations absdiff of sourcePixels is greater than pThreshold is true, the output pixel is set to pValue, otherwise it is set to absdiff of sourcePixels.

1.15.1.8 Image Threshold Less Than Value Greater Than Value Operations

1.15.1.8.1 Threshold Less Than Value Or Greater Than Value Operations

Replace image pixels less than thresholdLT or greater than thresholdGT with with valueLT or valueGT respectively.

1.15.1.8.1.1 Common parameters for nppiThreshold_LTValGTVal non-inplace and inplace functions:

param pSrcDst
In-Place Image Pointer for inplace functions.

param nSrcDstStep
In-Place-Image Line Step for inplace functions.

param pSrc
Source-Image Pointer for non-inplace functions.

param nSrcStep
Source-Image Line Step for non-inplace functions.

param pDst
Destination-Image Pointer for non-inplace functions.

param nDstStep
Destination-Image Line Step for non-inplace functions.

param oSizeROI
Region-Of-Interest (ROI).

param nThresholdLT
The thresholdLT value.

param nValueLT
The thresholdLT replacement value.

param nThresholdGT
The thresholdGT value.

param nValueGT
The thresholdGT replacement value.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes.

Functions

NppStatus nppiThreshold_LTVa1GTVa1_8u_C1R_Ctx(const *Npp8u* *pSrc, int nSrcStep, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp8u* nThresholdLT, const *Npp8u* nValueLT, const *Npp8u* nThresholdGT, const *Npp8u* nValueGT, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1GTVa1_8u_C1IR_Ctx(*Npp8u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp8u* nThresholdLT, const *Npp8u* nValueLT, const *Npp8u* nThresholdGT, const *Npp8u* nValueGT, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1GTVa1_16u_C1R_Ctx(const *Npp16u* *pSrc, int nSrcStep, *Npp16u* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16u* nThresholdLT, const *Npp16u* nValueLT, const *Npp16u* nThresholdGT, const *Npp16u* nValueGT, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVa1 non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1GTVa1_16u_C1IR_Ctx(*Npp16u* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16u* nThresholdLT, const *Npp16u* nValueLT, const *Npp16u* nThresholdGT, const *Npp16u* nValueGT, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1GTVal_16s_C1R_Ctx(const *Npp16s* *pSrc, int nSrcStep, *Npp16s* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp16s* nThresholdLT, const *Npp16s* nValueLT, const *Npp16s* nThresholdGT, const *Npp16s* nValueGT, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1GTVal_16s_C1IR_Ctx(*Npp16s* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp16s* nThresholdLT, const *Npp16s* nValueLT, const *Npp16s* nThresholdGT, const *Npp16s* nValueGT, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1GTVal_32f_C1R_Ctx(const *Npp32f* *pSrc, int nSrcStep, *Npp32f* *pDst, int nDstStep, *NppiSize* oSizeROI, const *Npp32f* nThresholdLT, const *Npp32f* nValueLT, const *Npp32f* nThresholdGT, const *Npp32f* nValueGT, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions*.

NppStatus nppiThreshold_LTVa1GTVal_32f_C1IR_Ctx(*Npp32f* *pSrcDst, int nSrcDstStep, *NppiSize* oSizeROI, const *Npp32f* nThresholdLT, const *Npp32f* nValueLT, const *Npp32f* nThresholdGT, const *Npp32f* nValueGT, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise

it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVal_8u_C3R_Ctx(const Npp8u *pSrc, int nSrcStep, Npp8u
                                                *pDst, int nDstStep, NppiSize oSizeROI,
                                                const Npp8u rThresholdsLT[3], const
                                                Npp8u rValuesLT[3], const Npp8u
                                                rThresholdsGT[3], const Npp8u
                                                rValuesGT[3], NppStreamContext
                                                nppStreamCtx)
```

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVal_8u_C3IR_Ctx(Npp8u *pSrcDst, int nSrcDstStep,
                                                NppiSize oSizeROI, const Npp8u
                                                rThresholdsLT[3], const Npp8u
                                                rValuesLT[3], const Npp8u
                                                rThresholdsGT[3], const Npp8u
                                                rValuesGT[3], NppStreamContext
                                                nppStreamCtx)
```

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVal_16u_C3R_Ctx(const Npp16u *pSrc, int nSrcStep, Npp16u
                                                *pDst, int nDstStep, NppiSize oSizeROI,
                                                const Npp16u rThresholdsLT[3], const
                                                Npp16u rValuesLT[3], const Npp16u
                                                rThresholdsGT[3], const Npp16u
                                                rValuesGT[3], NppStreamContext
                                                nppStreamCtx)
```

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVa1_16u_C3IR_Ctx(Npp16u *pSrcDst, int nSrcDstStep,  
                                                NppiSize oSizeROI, const Npp16u  
                                                rThresholdsLT[3], const Npp16u  
                                                rValuesLT[3], const Npp16u  
                                                rThresholdsGT[3], const Npp16u  
                                                rValuesGT[3], NppStreamContext  
                                                nppStreamCtx)
```

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVa1 non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVa1_16s_C3R_Ctx(const Npp16s *pSrc, int nSrcStep, Npp16s  
                                                *pDst, int nDstStep, NppiSize oSizeROI,  
                                                const Npp16s rThresholdsLT[3], const  
                                                Npp16s rValuesLT[3], const Npp16s  
                                                rThresholdsGT[3], const Npp16s  
                                                rValuesGT[3], NppStreamContext  
                                                nppStreamCtx)
```

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVa1 non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVa1_16s_C3IR_Ctx(Npp16s *pSrcDst, int nSrcDstStep,  
                                                NppiSize oSizeROI, const Npp16s  
                                                rThresholdsLT[3], const Npp16s  
                                                rValuesLT[3], const Npp16s  
                                                rThresholdsGT[3], const Npp16s  
                                                rValuesGT[3], NppStreamContext  
                                                nppStreamCtx)
```

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVa1 non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVa1_32f_C3R_Ctx(const Npp32f *pSrc, int nSrcStep, Npp32f  
                                                *pDst, int nDstStep, NppiSize oSizeROI,  
                                                const Npp32f rThresholdsLT[3], const  
                                                Npp32f rValuesLT[3], const Npp32f  
                                                rThresholdsGT[3], const Npp32f  
                                                rValuesGT[3], NppStreamContext  
                                                nppStreamCtx)
```

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVal_32f_C3IR_Ctx( Npp32f *pSrcDst, int nSrcDstStep,
                                                NppiSize oSizeROI, const Npp32f
                                                rThresholdsLT[3], const Npp32f
                                                rValuesLT[3], const Npp32f
                                                rThresholdsGT[3], const Npp32f
                                                rValuesGT[3], NppStreamContext
                                                nppStreamCtx)
```

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVal_8u_AC4R_Ctx( const Npp8u *pSrc, int nSrcStep, Npp8u
                                                *pDst, int nDstStep, NppiSize oSizeROI,
                                                const Npp8u rThresholdsLT[3], const
                                                Npp8u rValuesLT[3], const Npp8u
                                                rThresholdsGT[3], const Npp8u
                                                rValuesGT[3], NppStreamContext
                                                nppStreamCtx)
```

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVal_8u_AC4IR_Ctx( Npp8u *pSrcDst, int nSrcDstStep,
                                                NppiSize oSizeROI, const Npp8u
                                                rThresholdsLT[3], const Npp8u
                                                rValuesLT[3], const Npp8u
                                                rThresholdsGT[3], const Npp8u
                                                rValuesGT[3], NppStreamContext
                                                nppStreamCtx)
```

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVal non-inplace and inplace functions:*.

```
NppStatus nppiThreshold_LTVa1GTVa1_16u_AC4R_Ctx( const Npp16u *pSrc, int nSrcStep,
                                                    Npp16u *pDst, int nDstStep, NppiSize
                                                    oSizeROI, const Npp16u
                                                    rThresholdsLT[3], const Npp16u
                                                    rValuesLT[3], const Npp16u
                                                    rThresholdsGT[3], const Npp16u
                                                    rValuesGT[3], NppStreamContext
                                                    nppStreamCtx )
```

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVa1 non-inplace and inplace functions*.

```
NppStatus nppiThreshold_LTVa1GTVa1_16u_AC4IR_Ctx( Npp16u *pSrcDst, int nSrcDstStep,
                                                    NppiSize oSizeROI, const Npp16u
                                                    rThresholdsLT[3], const Npp16u
                                                    rValuesLT[3], const Npp16u
                                                    rThresholdsGT[3], const Npp16u
                                                    rValuesGT[3], NppStreamContext
                                                    nppStreamCtx )
```

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVa1 non-inplace and inplace functions*.

```
NppStatus nppiThreshold_LTVa1GTVa1_16s_AC4R_Ctx( const Npp16s *pSrc, int nSrcStep,
                                                    Npp16s *pDst, int nDstStep, NppiSize
                                                    oSizeROI, const Npp16s
                                                    rThresholdsLT[3], const Npp16s
                                                    rValuesLT[3], const Npp16s
                                                    rThresholdsGT[3], const Npp16s
                                                    rValuesGT[3], NppStreamContext
                                                    nppStreamCtx )
```

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVa1GTVa1 non-inplace and inplace functions*.

```
NppStatus nppiThreshold_LTVa1GTVa1_16s_AC4IR_Ctx( Npp16s *pSrcDst, int nSrcDstStep,
                                                    NppiSize oSizeROI, const Npp16s
                                                    rThresholdsLT[3], const Npp16s
                                                    rValuesLT[3], const Npp16s
                                                    rThresholdsGT[3], const Npp16s
                                                    rValuesGT[3], NppStreamContext
                                                    nppStreamCtx )
```

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVaGTVal non-inplace and inplace functions*.

```
NppStatus nppiThreshold_LTVaGTVal_32f_AC4R_Ctx( const Npp32f *pSrc, int nSrcStep,
                                                Npp32f *pDst, int nDstStep, NppiSize
                                                oSizeROI, const Npp32f
                                                rThresholdsLT[3], const Npp32f
                                                rValuesLT[3], const Npp32f
                                                rThresholdsGT[3], const Npp32f
                                                rValuesGT[3], NppStreamContext
                                                nppStreamCtx )
```

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVaGTVal non-inplace and inplace functions*.

```
NppStatus nppiThreshold_LTVaGTVal_32f_AC4IR_Ctx( Npp32f *pSrcDst, int nSrcDstStep,
                                                NppiSize oSizeROI, const Npp32f
                                                rThresholdsLT[3], const Npp32f
                                                rValuesLT[3], const Npp32f
                                                rThresholdsGT[3], const Npp32f
                                                rValuesGT[3], NppStreamContext
                                                nppStreamCtx )
```

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

For common parameter descriptions, see *Common parameters for nppiThreshold_LTVaGTVal non-inplace and inplace functions*.

1.15.2. Image Comparison Operations

1.15.2.1 Comparison Operations

Compare the pixels of two images or one image and a constant value and create a binary result image. In case of multi-channel image types, the condition must be fulfilled for all channels, otherwise the comparison is considered false. The “binary” result image is of type 8u_C1. False is represented by 0, true by NPP_MAX_8U.

1.15.2.2 Compare Images Operations

1.15.2.2.1 Compare Images Operations

Compare the pixels of two images and create a binary result image. In case of multi-channel image types, the condition must be fulfilled for all channels, otherwise the comparison is considered false. The “binary” result image is of type 8u_C1. False is represented by 0, true by NPP_MAX_8U.

1.15.2.2.1.1 Common parameters for nppiCompare functions:

param pSrc1
Source-Image Pointer.

param nSrc1Step
Source-Image Line Step.

param pSrc2
Source-Image Pointer.

param nSrc2Step
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param eComparisonOperation
Specifies the comparison operation to be used in the pixel comparison.

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiCompare_8u_C1R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions:*

NppStatus **nppiCompare_8u_C3R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

NppStatus **nppiCompare_8u_C4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

NppStatus **nppiCompare_8u_AC4R_Ctx**(const *Npp8u* *pSrc1, int nSrc1Step, const *Npp8u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

NppStatus **nppiCompare_16u_C1R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

NppStatus **nppiCompare_16u_C3R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

NppStatus **nppiCompare_16u_C4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

NppStatus **nppiCompare_16u_AC4R_Ctx**(const *Npp16u* *pSrc1, int nSrc1Step, const *Npp16u* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned short image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

```
NppStatus nppiCompare_16s_C1R_Ctx( const Npp16s *pSrc1, int nSrc1Step, const Npp16s *pSrc2,  
                                     int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize  
                                     oSizeROI, NppCmpOp eComparisonOperation,  
                                     NppStreamContext nppStreamCtx )
```

1 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

```
NppStatus nppiCompare_16s_C3R_Ctx( const Npp16s *pSrc1, int nSrc1Step, const Npp16s *pSrc2,  
                                     int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize  
                                     oSizeROI, NppCmpOp eComparisonOperation,  
                                     NppStreamContext nppStreamCtx )
```

3 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

```
NppStatus nppiCompare_16s_C4R_Ctx( const Npp16s *pSrc1, int nSrc1Step, const Npp16s *pSrc2,  
                                     int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize  
                                     oSizeROI, NppCmpOp eComparisonOperation,  
                                     NppStreamContext nppStreamCtx )
```

4 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

```
NppStatus nppiCompare_16s_AC4R_Ctx( const Npp16s *pSrc1, int nSrc1Step, const Npp16s  
                                     *pSrc2, int nSrc2Step, Npp8u *pDst, int nDstStep,  
                                     NppiSize oSizeROI, NppCmpOp eComparisonOperation,  
                                     NppStreamContext nppStreamCtx )
```

4 channel 16-bit signed short image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

```
NppStatus nppiCompare_32f_C1R_Ctx( const Npp32f *pSrc1, int nSrc1Step, const Npp32f *pSrc2,  
                                     int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize  
                                     oSizeROI, NppCmpOp eComparisonOperation,  
                                     NppStreamContext nppStreamCtx )
```

1 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions*.

```
NppStatus nppiCompare_32f_C3R_Ctx( const Npp32f *pSrc1, int nSrc1Step, const Npp32f *pSrc2,  
                                     int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize  
                                     oSizeROI, NppCmpOp eComparisonOperation,  
                                     NppStreamContext nppStreamCtx )
```


3 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions:*.

NppStatus **nppiCompare_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions:*.

NppStatus **nppiCompare_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed floating point compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

For common parameter descriptions, see *Common parameters for nppiCompare functions:*.

1.15.2.3 Compare Image With Constant Operations

1.15.2.3.1 Compare Image With Constant Operations

Compare the pixels of an image with a constant value and create a binary result image. In case of multi-channel image types, the condition must be fulfilled for all channels, otherwise the comparison is considered false. The "binary" result image is of type 8u_C1. False is represented by 0, true by NPP_MAX_8U.

1.15.2.3.1.1 Common parameters for nppiCompareC functions:

param pSrc

Source-Image Pointer.

param nSrcStep

Source-Image Line Step.

param nConstant

constant value for single channel functions.

param pConstants

pointer to a list of constant values, one per color channel for multi-channel functions.

param pDst

Destination-Image Pointer.

param nDstStep

Destination-Image Line Step.

param oSizeROI

Region-Of-Interest (ROI).

param eComparisonOperation

Specifies the comparison operation to be used in the pixel comparison.

param nppStreamCtx

Application Managed Stream Context.

return

Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiCompareC_8u_C1R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions:*.

NppStatus **nppiCompareC_8u_C3R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions:*.

NppStatus **nppiCompareC_8u_C4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions:*.

NppStatus **nppiCompareC_8u_AC4R_Ctx**(const *Npp8u* *pSrc, int nSrcStep, const *Npp8u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 8-bit unsigned char image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions:*.

NppStatus **nppiCompareC_16u_C1R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp16u* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit unsigned short image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions*.

NppStatus **nppiCompareC_16u_C3R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp16u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit unsigned short image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions*.

NppStatus **nppiCompareC_16u_C4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp16u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned short image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions*.

NppStatus **nppiCompareC_16u_AC4R_Ctx**(const *Npp16u* *pSrc, int nSrcStep, const *Npp16u* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit unsigned short image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions*.

NppStatus **nppiCompareC_16s_C1R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, const *Npp16s* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

1 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions*.

NppStatus **nppiCompareC_16s_C3R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, const *Npp16s* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

3 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions*.

NppStatus **nppiCompareC_16s_C4R_Ctx**(const *Npp16s* *pSrc, int nSrcStep, const *Npp16s* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *NppCmpOp* eComparisonOperation, *NppStreamContext* nppStreamCtx)

4 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions:*.

```
NppStatus nppiCompareC_16s_AC4R_Ctx(const Npp16s *pSrc, int nSrcStep, const Npp16s
                                     *pConstants, Npp8u *pDst, int nDstStep, NppiSize
                                     oSizeROI, NppCmpOp eComparisonOperation,
                                     NppStreamContext nppStreamCtx)
```

4 channel 16-bit signed short image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions:*.

```
NppStatus nppiCompareC_32f_C1R_Ctx(const Npp32f *pSrc, int nSrcStep, const Npp32f
                                     nConstant, Npp8u *pDst, int nDstStep, NppiSize oSizeROI,
                                     NppCmpOp eComparisonOperation, NppStreamContext
                                     nppStreamCtx)
```

1 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions:*.

```
NppStatus nppiCompareC_32f_C3R_Ctx(const Npp32f *pSrc, int nSrcStep, const Npp32f
                                     *pConstants, Npp8u *pDst, int nDstStep, NppiSize
                                     oSizeROI, NppCmpOp eComparisonOperation,
                                     NppStreamContext nppStreamCtx)
```

3 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions:*.

```
NppStatus nppiCompareC_32f_C4R_Ctx(const Npp32f *pSrc, int nSrcStep, const Npp32f
                                     *pConstants, Npp8u *pDst, int nDstStep, NppiSize
                                     oSizeROI, NppCmpOp eComparisonOperation,
                                     NppStreamContext nppStreamCtx)
```

4 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions:*.

```
NppStatus nppiCompareC_32f_AC4R_Ctx(const Npp32f *pSrc, int nSrcStep, const Npp32f
                                     *pConstants, Npp8u *pDst, int nDstStep, NppiSize
                                     oSizeROI, NppCmpOp eComparisonOperation,
                                     NppStreamContext nppStreamCtx)
```

4 channel 32-bit signed floating point compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

For common parameter descriptions, see *Common parameters for nppiCompareC functions:*.

1.15.2.4 Compare Image Differences With Epsilon Operations

1.15.2.4.1 Compare Image Differences With Epsilon Operations

Compare the pixels value differences of two images with an epsilon value and create a binary result image. In case of multi-channel image types, the condition must be fulfilled for all channels, otherwise the comparison is considered false. The “binary” result image is of type 8u_C1. False is represented by 0, true by NPP_MAX_8U.

1.15.2.4.1.1 Common parameters for nppiCompareEqualEps functions include:

param pSrc1
Source-Image Pointer.

param nSrc1Step
Source-Image Line Step.

param pSrc2
Source-Image Pointer.

param nSrc2Step
Source-Image Line Step.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param nEpsilon
epsilon tolerance value to compare to pixel absolute differences

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiCompareEqualEps_32f_C1R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nEpsilon, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

For common parameter descriptions, see *Common parameters for nppiCompareEqualEps functions include:*.

NppStatus **nppiCompareEqualEps_32f_C3R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nEpsilon, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

For common parameter descriptions, see *Common parameters for nppiCompareEqualEps functions include:*.

NppStatus **nppiCompareEqualEps_32f_C4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nEpsilon, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

For common parameter descriptions, see *Common parameters for nppiCompareEqualEps functions include:*.

NppStatus **nppiCompareEqualEps_32f_AC4R_Ctx**(const *Npp32f* *pSrc1, int nSrc1Step, const *Npp32f* *pSrc2, int nSrc2Step, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nEpsilon, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed floating point compare whether two images are equal within epsilon, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

For common parameter descriptions, see *Common parameters for nppiCompareEqualEps functions include:*.

1.15.2.5 Compare Image Difference To Constant With Epsilon Operations

1.15.2.5.1 Compare Image Difference With Constant Within Epsilon Operations

Compare differences between image pixels and constant within an epsilon value and create a binary result image. In case of multi-channel image types, the condition must be fulfilled for all channels, otherwise the comparison is considered false. The "binary" result image is of type 8u_C1. False is represented by 0, true by NPP_MAX_8U.

1.15.2.5.1.1 Common parameters for nppiCompareEqualEpsC functions:

param pSrc
Source-Image Pointer.

param nSrcStep
Source-Image Line Step.

param nConstant
 constant value for single channel functions.

param pConstants
 pointer to a list of constants, one per color channel for multi-channel image functions.

param pDst
Destination-Image Pointer.

param nDstStep
Destination-Image Line Step.

param oSizeROI
Region-Of-Interest (ROI).

param nEpsilon
 epsilon tolerance value to compare to per color channel pixel absolute differences

param nppStreamCtx
Application Managed Stream Context.

return
Image Data Related Error Codes, ROI Related Error Codes

Functions

NppStatus **nppiCompareEqualEpsC_32f_C1R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp32f* nConstant, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nEpsilon, *NppStreamContext* nppStreamCtx)

1 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

For common parameter descriptions, see *Common parameters for nppiCompareEqualEpsC functions:*.

NppStatus **nppiCompareEqualEpsC_32f_C3R_Ctx**(const *Npp32f* *pSrc, int nSrcStep, const *Npp32f* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nEpsilon, *NppStreamContext* nppStreamCtx)

3 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

For common parameter descriptions, see *Common parameters for nppiCompareEqualEpsC functions*.

NppStatus nppiCompareEqualEpsC_32f_C4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, const *Npp32f* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nEpsilon, *NppStreamContext* nppStreamCtx)

4 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

For common parameter descriptions, see *Common parameters for nppiCompareEqualEpsC functions*.

NppStatus nppiCompareEqualEpsC_32f_AC4R_Ctx(const *Npp32f* *pSrc, int nSrcStep, const *Npp32f* *pConstants, *Npp8u* *pDst, int nDstStep, *NppiSize* oSizeROI, *Npp32f* nEpsilon, *NppStreamContext* nppStreamCtx)

4 channel 32-bit signed floating point compare whether image and constant are equal within epsilon, not affecting Alpha.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

For common parameter descriptions, see *Common parameters for nppiCompareEqualEpsC functions*.

1.16. Image Memory Management Functions

Routines for allocating and deallocating pitched image storage.

These methods are provided for convenience. They allocate memory that may contain additional padding bytes at the end of each line of pixels. Though padding is not necessary for any of the NPP image-processing primitives to work correctly, its absence may cause severe performance degradation compared to properly padded images.

These functions can be found in the nppisu library. Linking to only the sub-libraries that you use can significantly save link time, application load time, and CUDA runtime startup time when using dynamic libraries.

Image Memory Allocation

ImageAllocator methods for 2D arrays of data.

The allocators have width and height parameters to specify the size of the image data being allocated. They return a pointer to the newly created memory and return the numbers of bytes between successive lines.

If the memory allocation failed due to lack of free device memory or device memory fragmentation the routine returns 0.

All allocators return memory with line strides that are beneficial for performance. It is not mandatory to use these allocators. Any valid CUDA device-memory pointers can be used by the NPP primitives and there are no restrictions on line strides.

Npp8u ***nppiMalloc_8u_C1**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

8-bit unsigned image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp8u ***nppiMalloc_8u_C2**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

2 channel 8-bit unsigned image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp8u ***nppiMalloc_8u_C3**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

3 channel 8-bit unsigned image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp8u ***nppiMalloc_8u_C4**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

4 channel 8-bit unsigned image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16u ***nppiMalloc_16u_C1**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

16-bit unsigned image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16u ***nppiMalloc_16u_C2**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

2 channel 16-bit unsigned image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16u ***nppiMalloc_16u_C3**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

3 channel 16-bit unsigned image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16u ***nppiMalloc_16u_C4**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

4 channel 16-bit unsigned image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16s ***nppiMalloc_16s_C1**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

16-bit signed image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16s ***nppiMalloc_16s_C2**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

2 channel 16-bit signed image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16s ***nppiMalloc_16s_C4**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

4 channel 16-bit signed image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16sc ***nppiMalloc_16sc_C1**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

1 channel 16-bit signed complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16sc ***nppiMalloc_16sc_C2**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

2 channel 16-bit signed complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16sc ***nppiMalloc_16sc_C3**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

3 channel 16-bit signed complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp16sc ***nppiMalloc_16sc_C4**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

4 channel 16-bit signed complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32s ***nppiMalloc_32s_C1**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

32-bit signed image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32s ***nppiMalloc_32s_C3**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

3 channel 32-bit signed image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32s *nppiMalloc_32s_C4(int nWidthPixels, int nHeightPixels, int *pStepBytes)

4 channel 32-bit signed image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32sc *nppiMalloc_32sc_C1(int nWidthPixels, int nHeightPixels, int *pStepBytes)

32-bit integer complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32sc *nppiMalloc_32sc_C2(int nWidthPixels, int nHeightPixels, int *pStepBytes)

2 channel 32-bit integer complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32sc *nppiMalloc_32sc_C3(int nWidthPixels, int nHeightPixels, int *pStepBytes)

3 channel 32-bit integer complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32sc *nppiMalloc_32sc_C4(int nWidthPixels, int nHeightPixels, int *pStepBytes)

4 channel 32-bit integer complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32f ***nppiMalloc_32f_C1**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

32-bit floating point image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32f ***nppiMalloc_32f_C2**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

2 channel 32-bit floating point image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32f ***nppiMalloc_32f_C3**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

3 channel 32-bit floating point image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32f ***nppiMalloc_32f_C4**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

4 channel 32-bit floating point image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32fc ***nppiMalloc_32fc_C1**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

32-bit float complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32fc ***nppiMalloc_32fc_C2**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

2 channel 32-bit float complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32fc ***nppiMalloc_32fc_C3**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

3 channel 32-bit float complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Npp32fc ***nppiMalloc_32fc_C4**(int nWidthPixels, int nHeightPixels, int *pStepBytes)

4 channel 32-bit float complex image memory allocator.

Parameters

- ▶ **nWidthPixels** – Image width.
- ▶ **nHeightPixels** – Image height.
- ▶ **pStepBytes** – *Line Step*.

Returns

Pointer to new image data.

Functions

void **nppiFree**(void *pData)

Free method for any 2D allocated memory.

This method should be used to free memory allocated with any of the nppiMalloc_<modifier> methods.

Parameters

- **pData** – A pointer to memory allocated using nppiMalloc_<modifier>.

1.17. Signal Arithmetic And Logical Operations

Functions that provide common arithmetic and logical operations.

1.17.1. Signal Arithmetic Functions

1.17.1.1 Arithmetic Operations

The set of arithmetic operations for signal processing available in the library.

1.17.1.2 Signal AddC

1.17.1.2.1 AddC

Adds a constant value to each sample of a signal.

Functions

NppStatus **nppsAddC_8u_ISfs_Ctx**(*Npp8u* nValue, *Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal add constant, scale, then clamp to saturated value

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be added to each vector element
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_8u_Sfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* nValue, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned charvector add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_16u_ISfs_Ctx**(*Npp16u* nValue, *Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_16u_Sfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* nValue, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short vector add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_16s_ISfs_Ctx**(*Npp16s* nValue, *Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_16s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* nValue, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_16sc_ISfs_Ctx**(*Npp16sc* nValue, *Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* nValue, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_32s_ISfs_Ctx**(*Npp32s* nValue, *Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer in place signal add constant and scale.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_32s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* nValue, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integers signal add constant and scale.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_32sc_ISfs_Ctx**(*Npp32sc* nValue, *Npp32sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add constant and scale.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_32sc_Sfs_Ctx**(const *Npp32sc* *pSrc, *Npp32sc* nValue, *Npp32sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal add constant and scale.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_32f_I_Ctx**(*Npp32f* nValue, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place signal add constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* nValue, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal add constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_32fc_I_Ctx**(*Npp32fc* nValue, *Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* nValue, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be added to each vector element.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_64f_I_Ctx**(*Npp64f* nValue, *Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point, in place signal add constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **nLength** – Length of the vectors, number of items.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* nValue, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating pointsignal add constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_64fc_I_Ctx**(*Npp64fc* nValue, *Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddC_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* nValue, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nValue** – Constant value to be added to each vector element
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.***1.17.1.3 Signal AddProductC****1.17.1.3.1 AddProductC**

Adds product of a constant and each sample of a source signal to the each sample of destination signal.

Functions

NppStatus **nppsAddProductC_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* nValue, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal add product of signal times constant to destination signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.***1.17.1.4 Signal MulC****1.17.1.4.1 MulC**

Multiplies each sample of a signal by a constant value.

Functions

NppStatus **nppsMulC_8u_ISfs_Ctx**(*Npp8u* nValue, *Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal times constant, scale, then clamp to saturated value

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_8u_Sfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* nValue, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal times constant, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be multiplied by each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_16u_ISfs_Ctx**(*Npp16u* nValue, *Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal times constant, scale, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be multiplied by each vector element
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_16u_Sfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* nValue, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal times constant, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be multiplied by each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMulC_16s_ISfs_Ctx**(*Npp16s* nValue, *Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal times constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMulC_16s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* nValue, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal times constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMulC_16sc_ISfs_Ctx**(*Npp16sc* nValue, *Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMulC_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* nValue, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_32s_ISfs_Ctx**(*Npp32s* nValue, *Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer in place signal times constant and scale.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_32s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* nValue, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal times constant and scale.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_32sc_ISfs_Ctx**(*Npp32sc* nValue, *Npp32sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal times constant and scale.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_32sc_Sfs_Ctx**(const *Npp32sc* *pSrc, *Npp32sc* nValue, *Npp32sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal times constant and scale.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_32f_I_Ctx**(*Npp32f* nValue, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place signal times constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* nValue, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal times constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_Low_32f16s_Ctx**(const *Npp32f* *pSrc, *Npp32f* nValue, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal times constant with output converted to 16-bit signed integer.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_32f16s_Sfs_Ctx**(const *Npp32f* *pSrc, *Npp32f* nValue, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit floating point signal times constant with output converted to 16-bit signed integer with scaling and saturation of output result.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_32fc_I_Ctx**(*Npp32fc* nValue, *Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal times constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be multiplied by each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* nValue, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal times constant.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be multiplied by each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_64f_I_Ctx**(*Npp64f* nValue, *Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point, in place signal times constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be multiplied by each vector element
- **nLength** – Length of the vectors, number of items.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* nValue, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal times constant.

Parameters

- **pSrc** – *Source Signal Pointer.*

- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_64f64s_ISfs_Ctx**(*Npp64f* nValue, *Npp64s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit floating point signal times constant with in place conversion to 64-bit signed integer and with scaling and saturation of output result.

Parameters

- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_64fc_I_Ctx**(*Npp64fc* nValue, *Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal times constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMulC_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* nValue, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal times constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nValue** – Constant value to be multiplied by each vector element
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.***1.17.1.5 Signal SubC****1.17.1.5.1 SubC**

Subtracts a constant from each sample of a signal.

Functions

NppStatus **nppsSubC_8u_ISfs_Ctx**(*Npp8u* nValue, *Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal subtract constant, scale, then clamp to saturated value

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be subtracted from each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsSubC_8u_Sfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* nValue, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be subtracted from each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsSubC_16u_ISfs_Ctx**(*Npp16u* nValue, *Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nValue** – Constant value to be subtracted from each vector element
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_16u_Sfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* nValue, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nValue** – Constant value to be subtracted from each vector element
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_16s_ISfs_Ctx**(*Npp16s* nValue, *Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nValue** – Constant value to be subtracted from each vector element
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_16s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* nValue, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal subtract constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nValue** – Constant value to be subtracted from each vector element

- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_16sc_ISfs_Ctx**(*Npp16sc* nValue, *Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be subtracted from each vector element
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* nValue, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be subtracted from each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_32s_ISfs_Ctx**(*Npp32s* nValue, *Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer in place signal subtract constant and scale.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*

- **nValue** – Constant value to be subtracted from each vector element
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_32s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* nValue, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal subtract constant and scale.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be subtracted from each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_32sc_ISfs_Ctx**(*Npp32sc* nValue, *Npp32sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract constant and scale.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be subtracted from each vector element
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_32sc_Sfs_Ctx**(const *Npp32sc* *pSrc, *Npp32sc* nValue, *Npp32sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract constant and scale.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be subtracted from each vector element

- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_32f_I_Ctx**(*Npp32f* nValue, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place signal subtract constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be subtracted from each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* nValue, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal subtract constant.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be subtracted from each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_32fc_I_Ctx**(*Npp32fc* nValue, *Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be subtracted from each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* nValue, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be subtracted from each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_64f_I_Ctx**(*Npp64f* nValue, *Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point, in place signal subtract constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be subtracted from each vector element
- ▶ **nLength** – Length of the vectors, number of items.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* nValue, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal subtract constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be subtracted from each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_64fc_I_Ctx**(*Npp64fc* nValue, *Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be subtracted from each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubC_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* nValue, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be subtracted from each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.6 Signal SubCRev**1.17.1.6.1 SubCRev**

Subtracts each sample of a signal from a constant.

Functions

NppStatus **nppsSubCRev_8u_ISfs_Ctx**(*Npp8u* nValue, *Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal subtract from constant, scale, then clamp to saturated value

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_8u_Sfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* nValue, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal subtract from constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_16u_ISfs_Ctx**(*Npp16u* nValue, *Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal subtract from constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_16u_Sfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* nValue, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal subtract from constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_16s_ISfs_Ctx**(*Npp16s* nValue, *Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal subtract from constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_16s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* nValue, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal subtract from constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_16sc_ISfs_Ctx**(*Npp16sc* nValue, *Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* nValue, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_32s_ISfs_Ctx**(*Npp32s* nValue, *Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer in place signal subtract from constant and scale.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_32s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* nValue, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integersignal subtract from constant and scale.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_32sc_ISfs_Ctx**(*Npp32sc* nValue, *Npp32sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant and scale.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_32sc_Sfs_Ctx**(const *Npp32sc* *pSrc, *Npp32sc* nValue, *Npp32sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract from constant and scale.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_32f_I_Ctx**(*Npp32f* nValue, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place signal subtract from constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* nValue, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal subtract from constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_32fc_I_Ctx**(*Npp32fc* nValue, *Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* nValue, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract from constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_64f_I_Ctx**(*Npp64f* nValue, *Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point, in place signal subtract from constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **nLength** – Length of the vectors, number of items.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* nValue, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal subtract from constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_64fc_I_Ctx**(*Npp64fc* nValue, *Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract from constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSubCRev_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* nValue, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract from constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value each vector element is to be subtracted from
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.7 Signal DivC**1.17.1.7.1 DivC**

Divides each sample of a signal by a constant.

Functions

NppStatus **nppsDivC_8u_ISfs_Ctx**(*Npp8u* nValue, *Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal divided by constant, scale, then clamp to saturated value

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be divided into each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_8u_Sfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* nValue, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be divided into each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_16u_ISfs_Ctx**(*Npp16u* nValue, *Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be divided into each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_16u_Sfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* nValue, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be divided into each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_16s_ISfs_Ctx**(*Npp16s* nValue, *Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be divided into each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_16s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* nValue, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal divided by constant, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be divided into each vector element

- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_16sc_ISfs_Ctx**(*Npp16sc* nValue, *Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be divided into each vector element
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* nValue, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be divided into each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_32f_I_Ctx**(*Npp32f* nValue, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place signal divided by constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*

- **nValue** – Constant value to be divided into each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* nValue, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal divided by constant.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be divided into each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_32fc_I_Ctx**(*Npp32fc* nValue, *Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal divided by constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be divided into each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* nValue, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal divided by constant.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be divided into each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_64f_I_Ctx**(*Npp64f* nValue, *Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point in place signal divided by constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be divided into each vector element
- ▶ **nLength** – Length of the vectors, number of items.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* nValue, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal divided by constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be divided into each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_64fc_I_Ctx**(*Npp64fc* nValue, *Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal divided by constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be divided into each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivC_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* nValue, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal divided by constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be divided into each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.8 Signal DivCRev**1.17.1.8.1 DivCRev**

Divides a constant by each sample of a signal.

Functions

NppStatus **nppsDivCRev_16u_I_Ctx**(*Npp16u* nValue, *Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place constant divided by signal, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be divided by each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivCRev_16u_Ctx**(const *Npp16u* *pSrc, *Npp16u* nValue, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal divided by constant, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be divided by each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivCRev_32f_I_Ctx**(*Npp32f* nValue, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place constant divided by signal.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nValue** – Constant value to be divided by each vector element
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDivCRev_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* nValue, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point constant divided by signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nValue** – Constant value to be divided by each vector element
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.9 Signal Add

1.17.1.9.1 Add

Sample by sample addition of two signals.

Functions

NppStatus **nppsAdd_16s_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short signal add signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_16u_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal add signal, then clamp to saturated value.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer.* signal2 elements to be added to signal1 elements
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_32u_Ctx**(const *Npp32u* *pSrc1, const *Npp32u* *pSrc2, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned int signal add signal, then clamp to saturated value.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer.* signal2 elements to be added to signal1 elements
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal add signal, then clamp to saturated value.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer.* signal2 elements to be added to signal1 elements
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal add signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_32fc_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point signal add signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_64fc_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point signal add signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_8u16u_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal add signal with 16-bit unsigned result, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.* signal2 elements to be added to signal1 elements
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_16s32f_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short signal add signal with 32-bit floating point result, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.* signal2 elements to be added to signal1 elements
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_8u_Sfs_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char add signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer,* signal2 elements to be added to signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_16u_Sfs_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short add signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be added to signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_16s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short add signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be added to signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_32s_Sfs_Ctx**(const *Npp32s* *pSrc1, const *Npp32s* *pSrc2, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer add signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be added to signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_64s_Sfs_Ctx**(const *Npp64s* *pSrc1, const *Npp64s* *pSrc2, *Npp64s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit signed integer add signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be added to signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed complex short add signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be added to signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_32sc_Sfs_Ctx**(const *Npp32sc* *pSrc1, const *Npp32sc* *pSrc2, *Npp32sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed complex integer add signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be added to signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.

- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_16s_I_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal add signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_32f_I_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place signal add signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_64f_I_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point in place signal add signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_32fc_I_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point in place signal add signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_64fc_I_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point in place signal add signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_16s32s_I_Ctx**(const *Npp16s* *pSrc, *Npp32s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16/32-bit signed short in place signal add signal with 32-bit signed integer results, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_8u_ISfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal add signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_16u_ISfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal add signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_16s_ISfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal add signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_32s_ISfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer in place signal add signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.

- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_16sc_ISfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit complex signed short in place signal add signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAdd_32sc_ISfs_Ctx**(const *Npp32sc* *pSrc, *Npp32sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit complex signed integer in place signal add signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be added to signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.10 Signal AddProduct

1.17.1.10.1 AddProduct

Adds sample by sample product of two signals to the destination signal.

Functions

NppStatus **nppsAddProduct_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **pDst** – *Destination Signal Pointer*. product of source1 and source2 signal elements to be added to destination elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddProduct_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **pDst** – *Destination Signal Pointer*. product of source1 and source2 signal elements to be added to destination elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddProduct_32fc_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.* product of source1 and source2 signal elements to be added to destination elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddProduct_64fc_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.* product of source1 and source2 signal elements to be added to destination elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddProduct_16s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.* product of source1 and source2 signal elements to be added to destination elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddProduct_32s_Sfs_Ctx**(const *Npp32s* *pSrc1, const *Npp32s* *pSrc2, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.* product of source1 and source2 signal elements to be added to destination elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAddProduct_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal add product of source signal1 times source signal2 to 32-bit signed integer destination signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.* product of source1 and source2 signal elements to be added to destination elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.11 Signal Mul

1.17.1.11.1 Mul

Sample by sample multiplication the samples of two signals.

Functions

NppStatus **nppsMul_16s_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short signal times signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal times signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal times signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32fc_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point signal times signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_64fc_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point signal times signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_8u16u_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal times signal with 16-bit unsigned result, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_16s32f_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short signal times signal with 32-bit floating point result, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32f32fc_Ctx**(const *Npp32f* *pSrc1, const *Npp32fc* *pSrc2, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal times 32-bit complex floating point signal with complex 32-bit floating point result, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_8u_Sfs_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal times signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be multiplied by signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_16u_Sfs_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal times signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be multiplied by signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_16s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal times signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be multiplied by signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32s_Sfs_Ctx**(const *Npp32s* *pSrc1, const *Npp32s* *pSrc2, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal times signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be multiplied by signal1 elements.

- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed complex short signal times signal, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer*, signal2 elements to be multiplied by signal1 elements.
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32sc_Sfs_Ctx**(const *Npp32sc* *pSrc1, const *Npp32sc* *pSrc2, *Npp32sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed complex integer signal times signal, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer*, signal2 elements to be multiplied by signal1 elements.
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_16u16s_Sfs_Ctx**(const *Npp16u* *pSrc1, const *Npp16s* *pSrc2, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal times 16-bit signed short signal, scale, then clamp to 16-bit signed saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be multiplied by signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal times signal, scale, then clamp to 32-bit signed saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be multiplied by signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32s32sc_Sfs_Ctx**(const *Npp32s* *pSrc1, const *Npp32sc* *pSrc2, *Npp32sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal times 32-bit complex signed integer signal, scale, then clamp to 32-bit complex integer saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be multiplied by signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_Low_32s_Sfs_Ctx**(const *Npp32s* *pSrc1, const *Npp32s* *pSrc2, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal times signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer*, signal2 elements to be multiplied by signal1 elements.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_16s_I_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal times signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be multiplied by signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32f_I_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place signal times signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be multiplied by signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_64f_I_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point in place signal times signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32fc_I_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point in place signal times signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_64fc_I_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point in place signal times signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32f32fc_I_Ctx**(const *Npp32f* *pSrc, *Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point in place signal times 32-bit floating point signal, then clamp to 32-bit complex floating point saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be multiplied by signal1 elements
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_8u_ISfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal times signal, with scaling, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be multiplied by signal1 elements
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_16u_ISfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal times signal, with scaling, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be multiplied by signal1 elements
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_16s_ISfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal times signal, with scaling, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be multiplied by signal1 elements

- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32s_ISfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer in place signal times signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_16sc_ISfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit complex signed short in place signal times signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be multiplied by signal1 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32sc_ISfs_Ctx**(const *Npp32sc* *pSrc, *Npp32sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit complex signed integer in place signal times signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal2 elements to be multiplied by signal1 elements

- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMul_32s32sc_ISfs_Ctx**(const *Npp32s* *pSrc, *Npp32sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit complex signed integer in place signal times 32-bit signed integer signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be multiplied by signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.12 Signal Sub**1.17.1.12.1 Sub**

Sample by sample subtraction of the samples of two signals.

Functions

NppStatus **nppsSub_16s_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short signal subtract signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.* signal1 elements to be subtracted from signal2 elements
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal subtract signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal subtract signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_32fc_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point signal subtract signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_64fc_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point signal subtract signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_16s32f_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short signal subtract 16-bit signed short signal, then clamp and convert to 32-bit floating point saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_8u_Sfs_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal subtract signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 elements to be subtracted from signal2 elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_16u_Sfs_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal subtract signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 elements to be subtracted from signal2 elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_16s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal subtract signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 elements to be subtracted from signal2 elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_32s_Sfs_Ctx**(const *Npp32s* *pSrc1, const *Npp32s* *pSrc2, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal subtract signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 elements to be subtracted from signal2 elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.

- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed complex short signal subtract signal, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer*, signal1 elements to be subtracted from signal2 elements.
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_32sc_Sfs_Ctx**(const *Npp32sc* *pSrc1, const *Npp32sc* *pSrc2, *Npp32sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed complex integer signal subtract signal, scale, then clamp to saturated value.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer*, signal1 elements to be subtracted from signal2 elements.
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_16s_I_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal subtract signal, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*

- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_32f_I_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place signal subtract signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_64f_I_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point in place signal subtract signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_32fc_I_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point in place signal subtract signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsSub_64fc_I_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point in place signal subtract signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsSub_8u_ISfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsSub_16u_ISfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsSub_16s_ISfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_32s_ISfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_16sc_ISfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit complex signed short in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSub_32sc_ISfs_Ctx**(const *Npp32sc* *pSrc, *Npp32sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit complex signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 elements to be subtracted from signal2 elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.13 Signal Div

1.17.1.13.1 Div

Sample by sample division of the samples of two signals.

Functions

NppStatus **nppsDiv_8u_Sfs_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal divide signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_16u_Sfs_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal divide signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_16s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal divide signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_32s_Sfs_Ctx**(const *Npp32s* *pSrc1, const *Npp32s* *pSrc2, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal divide signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed complex short signal divide signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_32s16s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp32s* *pSrc2, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal divided by 16-bit signed short signal, scale, then clamp to 16-bit signed short saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal divide signal, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal divide signal, then clamp to saturated value.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_32fc_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point signal divide signal, then clamp to saturated value.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_64fc_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point signal divide signal, then clamp to saturated value.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsDiv_8u_ISfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal divide signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsDiv_16u_ISfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsDiv_16s_ISfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal divide signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsDiv_16sc_ISfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit complex signed short in place signal divide signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_32s_ISfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer in place signal divide signal, with scaling, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_32f_I_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place signal divide signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_64f_I_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point in place signal divide signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_32fc_I_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point in place signal divide signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_64fc_I_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point in place signal divide signal, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.14 Signal Div Round

1.17.1.14.1 Div_Round

Sample by sample division of the samples of two signals with rounding.

Functions

NppStatus **nppsDiv_Round_8u_Sfs_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, *Npp8u* *pDst, size_t nLength, *NppRoundMode* nRndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal divide signal, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nRndMode** – various rounding modes.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_Round_16u_Sfs_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, *Npp16u* *pDst, size_t nLength, *NppRoundMode* nRndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal divide signal, scale, round, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nRndMode** – various rounding modes.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_Round_16s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, *Npp16s* *pDst, size_t nLength, *NppRoundMode* nRndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal divide signal, scale, round, then clamp to saturated value.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*, signal1 divisor elements to be divided into signal2 dividend elements.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nRndMode** – various rounding modes.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_Round_8u_ISfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pSrcDst, size_t nLength, *NppRoundMode* nRndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal divide signal, with scaling, rounding then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nRndMode** – various rounding modes.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_Round_16u_ISfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pSrcDst, size_t nLength, *NppRoundMode* nRndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal divide signal, with scaling, rounding then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 divisor elements to be divided into signal2 dividend elements

- ▶ **nLength** – *Signal Length*.
- ▶ **nRndMode** – various rounding modes.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDiv_Round_16s_ISfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pSrcDst, size_t nLength, *NppRoundMode* nRndMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal divide signal, with scaling, rounding then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pSrcDst** – *In-Place Signal Pointer*. signal1 divisor elements to be divided into signal2 dividend elements
- ▶ **nLength** – *Signal Length*.
- ▶ **nRndMode** – various rounding modes.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.15 Signal Abs

1.17.1.15.1 Abs

Absolute value of each sample of a signal.

Functions

NppStatus **nppsAbs_16s_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short signal absolute value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsAbs_32s_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

32-bit signed integer signal absolute value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsAbs_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

32-bit floating point signal absolute value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsAbs_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

64-bit floating point signal absolute value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsAbs_16s_I_Ctx**(*Npp16s* *pSrcDst, size_t nLength, *NppStreamContext*
nppStreamCtx)

16-bit signed short signal absolute value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*

- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAbs_32s_I_Ctx**(*Npp32s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal absolute value.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAbs_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal absolute value.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAbs_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal absolute value.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.16 Signal Square

1.17.1.16.1 Sqr

Squares each sample of a signal.

Functions

NppStatus **nppsSqr_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal squared.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal squared.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point signal squared.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point signal squared.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal squared.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal squared.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_32fc_I_Ctx**(*Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point signal squared.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_64fc_I_Ctx**(*Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point signal squared.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_8u_Sfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal squared, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_16u_Sfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal squared, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_16s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal squared, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit complex signed short signal squared, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_8u_ISfs_Ctx**(*Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal squared, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_16u_ISfs_Ctx**(*Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal squared, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_16s_ISfs_Ctx**(*Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal squared, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqr_16sc_ISfs_Ctx**(*Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit complex signed short signal squared, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.17 Signal Square Root

1.17.1.17.1 Sqrt

Square root of each sample of a signal.

Functions

NppStatus **nppsSqrt_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal square root.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

64-bit floating point signal square root.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

32-bit complex floating point signal square root.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

64-bit complex floating point signal square root.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *NppStreamContext*
nppStreamCtx)

32-bit floating point signal square root.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*

- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal square root.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_32fc_I_Ctx**(*Npp32fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex floating point signal square root.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_64fc_I_Ctx**(*Npp64fc* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex floating point signal square root.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_8u_Sfs_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal square root, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*

- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_16u_Sfs_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal square root, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_16s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal square root, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit complex signed short signal square root, scale, then clamp to saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsSqrt_64s_Sfs_Ctx**(const *Npp64s* *pSrc, *Npp64s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit signed integer signal square root, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsSqrt_32s16s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsSqrt_64s16s_Sfs_Ctx**(const *Npp64s* *pSrc, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsSqrt_8u_ISfs_Ctx**(*Npp8u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal square root, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_16u_ISfs_Ctx**(*Npp16u* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal square root, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_16s_ISfs_Ctx**(*Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal square root, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_16sc_ISfs_Ctx**(*Npp16sc* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit complex signed short signal square root, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*

- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSqrt_64s_ISfs_Ctx**(*Npp64s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit signed integer signal square root, scale, then clamp to saturated value.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.18 Signal Cube Root**1.17.1.18.1 Cubrt**

Cube root of each sample of a signal.

Functions

NppStatus **nppsCubrt_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal cube root.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCubrt_32s16s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal cube root, scale, then clamp to 16-bit signed integer saturated value.

Parameters

- **pSrc** – *Source Signal Pointer.*

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.19 Signal Exp

1.17.1.19.1 Exp

E raised to the power of each sample of a signal.

Functions

NppStatus **nppsExp_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

32-bit floating point signal exponent.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsExp_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

64-bit floating point signal exponent.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsExp_32f64f_Ctx**(const *Npp32f* *pSrc, *Npp64f* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

32-bit floating point signal exponent with 64-bit floating point result.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsExp_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal exponent.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsExp_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal exponent.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsExp_16s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal exponent, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsExp_32s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal exponent, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsExp_64s_Sfs_Ctx**(const *Npp64s* *pSrc, *Npp64s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit signed integer signal exponent, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsExp_16s_ISfs_Ctx**(*Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal exponent, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsExp_32s_ISfs_Ctx**(*Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal exponent, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsExp_64s_ISfs_Ctx**(*Npp64s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit signed integer signal exponent, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.20 Signal Ln**1.17.1.20.1 Ln**

Natural logarithm of each sample of a signal.

Functions

NppStatus **nppsLn_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal natural logarithm.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLn_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal natural logarithm.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLn_64f32f_Ctx**(const *Npp64f* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal natural logarithm with 32-bit floating point result.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLn_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal natural logarithm.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLn_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal natural logarithm.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLn_16s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal natural logarithm, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLn_32s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLn_32s16s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal natural logarithm, scale, then clamp to 16-bit signed short saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLn_16s_ISfs_Ctx**(*Npp16s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal natural logarithm, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLn_32s_ISfs_Ctx**(*Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.21 Signal 10Log10

1.17.1.21.1 10Log10

Ten times the decimal logarithm of each sample of a signal.

Functions

NppStatus **npps10Log10_32s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **npps10Log10_32s_ISfs_Ctx**(*Npp32s* *pSrcDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.22 Signal SumLn

1.17.1.22.1 SumLn

Sums up the natural logarithm of each sample of a signal.

Functions

NppStatus **nppsSumLnGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for 32f SumLn.

This primitive provides the correct buffer size for nppsSumLn_32f.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsSumLn_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit floating point signal sum natural logarithm.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSumLnGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for 64f SumLn.

This primitive provides the correct buffer size for nppsSumLn_64f.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsSumLn_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit floating point signal sum natural logarithm.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nLength** – *Signal Length.*
- **pDst** – Pointer to the output result.
- **pDeviceBuffer** – Pointer to the required device memory allocation.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSumLnGetBufferSize_32f64f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for 32f64f SumLn.

This primitive provides the correct buffer size for nppsSumLn_32f64f.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsSumLn_32f64f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit floating point input, 64-bit floating point output signal sum natural logarithm.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSumLnGetBufferSize_16s32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for 16s32f SumLn.

This primitive provides the correct buffer size for nppsSumLn_16s32f.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsSumLn_16s32f_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer input, 32-bit floating point output signal sum natural logarithm.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.23 Signal ArcTan

1.17.1.23.1 Arctan

Inverse tangent of each sample of a signal.

Functions

NppStatus **nppsArctan_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal inverse tangent.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsArctan_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal inverse tangent.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsArctan_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point signal inverse tangent.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsArctan_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point signal inverse tangent.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.24 Signal Normalize

1.17.1.24.1 Normalize

Normalize each sample of a real or complex signal using offset and division operations.

Functions

NppStatus **nppsNormalize_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *Npp32f* vSub, *Npp32f* vDiv, *NppStreamContext* nppStreamCtx)

32-bit floating point signal normalize.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **vSub** – value subtracted from each signal element before division
- ▶ **vDiv** – divisor of post-subtracted signal element dividend
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormalize_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pDst, size_t nLength, *Npp32fc* vSub, *Npp32f* vDiv, *NppStreamContext* nppStreamCtx)

32-bit complex floating point signal normalize.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*

- ▶ **vSub** – value subtracted from each signal element before division
- ▶ **vDiv** – divisor of post-subtracted signal element dividend
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormalize_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength, *Npp64f* vSub, *Npp64f* vDiv, *NppStreamContext* nppStreamCtx)

64-bit floating point signal normalize.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **vSub** – value subtracted from each signal element before division
- ▶ **vDiv** – divisor of post-subtracted signal element dividend
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormalize_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pDst, size_t nLength, *Npp64fc* vSub, *Npp64f* vDiv, *NppStreamContext* nppStreamCtx)

64-bit complex floating point signal normalize.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **vSub** – value subtracted from each signal element before division
- ▶ **vDiv** – divisor of post-subtracted signal element dividend
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormalize_16s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, *Npp16s* vSub, int vDiv, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short signal normalize, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pDst** – *Destination Signal Pointer*.

- ▶ **nLength** – *Signal Length.*
- ▶ **vSub** – value subtracted from each signal element before division
- ▶ **vDiv** – divisor of post-subtracted signal element dividend
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormalize_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pDst, size_t nLength, *Npp16sc* vSub, int vDiv, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit complex signed short signal normalize, scale, then clamp to saturated value.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **vSub** – value subtracted from each signal element before division
- ▶ **vDiv** – divisor of post-subtracted signal element dividend
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.1.25 Signal Cauchy, CauchyD, And CauchyDD2

1.17.1.25.1 Cauchy, CauchyD, and CauchyDD2

Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.

Functions

NppStatus **nppsCauchy_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *Npp32f* nParam, *NppStreamContext* nppStreamCtx)

32-bit floating point signal Cauchy error calculation.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nParam** – constant used in Cauchy formula
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCauchyD_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *Npp32f* nParam, *NppStreamContext* nppStreamCtx)

32-bit floating point signal Cauchy first derivative.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nParam** – constant used in Cauchy formula
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCauchyDD2_32f_I_Ctx**(*Npp32f* *pSrcDst, *Npp32f* *pD2FVal, size_t nLength, *Npp32f* nParam, *NppStreamContext* nppStreamCtx)

32-bit floating point signal Cauchy first and second derivatives.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **pD2FVal** – *Source Signal Pointer.* This signal contains the second derivative of the source signal.
- ▶ **nLength** – *Signal Length.*
- ▶ **nParam** – constant used in Cauchy formula
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.2. Logical And Shift Operations

1.17.2.1 Logical And Shift Operations

The set of logical and shift operations for signal processing available in the library.

1.17.2.2 Signal AndC

1.17.2.2.1 AndC

Bitwise AND of a constant and each sample of a signal.

Functions

NppStatus **nppsAndC_8u_Ctx**(const *Npp8u* *pSrc, *Npp8u* nValue, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal and with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be anded with each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAndC_16u_Ctx**(const *Npp16u* *pSrc, *Npp16u* nValue, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal and with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be anded with each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAndC_32u_Ctx**(const *Npp32u* *pSrc, *Npp32u* nValue, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer signal and with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be anded with each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAndC_8u_I_Ctx**(*Npp8u* nValue, *Npp8u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal and with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be added with each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAndC_16u_I_Ctx**(*Npp16u* nValue, *Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal and with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be added with each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAndC_32u_I_Ctx**(*Npp32u* nValue, *Npp32u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned signed integer in place signal and with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be added with each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.2.3 Signal And

1.17.2.3.1 And

Sample by sample bitwise AND of samples from two signals.

Functions

NppStatus **nppsAnd_8u_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal and with signal.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be anded with signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAnd_16u_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal and with signal.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be anded with signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAnd_32u_Ctx**(const *Npp32u* *pSrc1, const *Npp32u* *pSrc2, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer signal and with signal.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be anded with signal1 elements

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAnd_8u_I_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal and with signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be anded with signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAnd_16u_I_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal and with signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be anded with signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAnd_32u_I_Ctx**(const *Npp32u* *pSrc, *Npp32u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer in place signal and with signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be anded with signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.2.4 Signal OrC

1.17.2.4.1 OrC

Bitwise OR of a constant and each sample of a signal.

Functions

NppStatus **nppsOrC_8u_Ctx**(const *Npp8u* *pSrc, *Npp8u* nValue, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal or with constant.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be ored with each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsOrC_16u_Ctx**(const *Npp16u* *pSrc, *Npp16u* nValue, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal or with constant.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be ored with each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsOrC_32u_Ctx**(const *Npp32u* *pSrc, *Npp32u* nValue, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer signal or with constant.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be ored with each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsOrC_8u_I_Ctx**(*Npp8u* nValue, *Npp8u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal or with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be ored with each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsOrC_16u_I_Ctx**(*Npp16u* nValue, *Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal or with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be ored with each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsOrC_32u_I_Ctx**(*Npp32u* nValue, *Npp32u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned signed integer in place signal or with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be ored with each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.2.5 Signal Or

1.17.2.5.1 Or

Sample by sample bitwise OR of the samples from two signals.

Functions

NppStatus **nppsOr_8u_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal or with signal.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be ored with signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsOr_16u_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal or with signal.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be ored with signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsOr_32u_Ctx**(const *Npp32u* *pSrc1, const *Npp32u* *pSrc2, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer signal or with signal.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be ored with signal1 elements

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsOr_8u_I_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal or with signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be ored with signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsOr_16u_I_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal or with signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be ored with signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsOr_32u_I_Ctx**(const *Npp32u* *pSrc, *Npp32u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer in place signal or with signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be ored with signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.2.6 Signal XorC

1.17.2.6.1 XorC

Bitwise XOR of a constant and each sample of a signal.

Functions

NppStatus **nppsXorC_8u_Ctx**(const *Npp8u* *pSrc, *Npp8u* nValue, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal exclusive or with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be exclusive ored with each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsXorC_16u_Ctx**(const *Npp16u* *pSrc, *Npp16u* nValue, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal exclusive or with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be exclusive ored with each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsXorC_32u_Ctx**(const *Npp32u* *pSrc, *Npp32u* nValue, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer signal exclusive or with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be exclusive ored with each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsXorC_8u_I_Ctx**(*Npp8u* nValue, *Npp8u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal exclusive or with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be exclusive ored with each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsXorC_16u_I_Ctx**(*Npp16u* nValue, *Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal exclusive or with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be exclusive ored with each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsXorC_32u_I_Ctx**(*Npp32u* nValue, *Npp32u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned signed integer in place signal exclusive or with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be exclusive ored with each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.2.7 Signal Xor

1.17.2.7.1 Xor

Sample by sample bitwise XOR of the samples from two signals.

Functions

NppStatus **nppsXor_8u_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal exclusive or with signal.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be exclusive ored with signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsXor_16u_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal exclusive or with signal.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be exclusive ored with signal1 elements
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsXor_32u_Ctx**(const *Npp32u* *pSrc1, const *Npp32u* *pSrc2, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer signal exclusive or with signal.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*. signal2 elements to be exclusive ored with signal1 elements

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsXor_8u_I_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pSrcDst, size_t nLength,
NppStreamContext nppStreamCtx)

8-bit unsigned char in place signal exclusive or with signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be exclusive ored with signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsXor_16u_I_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pSrcDst, size_t nLength,
NppStreamContext nppStreamCtx)

16-bit unsigned short in place signal exclusive or with signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be exclusive ored with signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsXor_32u_I_Ctx**(const *Npp32u* *pSrc, *Npp32u* *pSrcDst, size_t nLength,
NppStreamContext nppStreamCtx)

32-bit unsigned integer in place signal exclusive or with signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.* signal2 elements to be exclusive ored with signal1 elements
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.2.8 Signal Not

1.17.2.8.1 Not

Bitwise NOT of each sample of a signal.

Functions

NppStatus **nppsNot_8u_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char not signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNot_16u_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short not signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNot_32u_Ctx**(const *Npp32u* *pSrc, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer not signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNot_8u_I_Ctx**(*Npp8u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place not signal.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNot_16u_I_Ctx**(*Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place not signal.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNot_32u_I_Ctx**(*Npp32u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned signed integer in place not signal.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.2.9 Signal LShiftC

1.17.2.9.1 LShiftC

Left shifts the bits of each sample of a signal by a constant amount.

Functions

NppStatus **nppsLShiftC_8u_Ctx**(const *Npp8u* *pSrc, int nValue, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal left shift with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be used to left shift each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLShiftC_16u_Ctx**(const *Npp16u* *pSrc, int nValue, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal left shift with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be used to left shift each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLShiftC_16s_Ctx**(const *Npp16s* *pSrc, int nValue, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short signal left shift with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be used to left shift each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLShiftC_32u_Ctx**(const *Npp32u* *pSrc, int nValue, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer signal left shift with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be used to left shift each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLShiftC_32s_Ctx**(const *Npp32s* *pSrc, int nValue, *Npp32s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal left shift with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be used to left shift each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLShiftC_8u_I_Ctx**(int nValue, *Npp8u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal left shift with constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be used to left shift each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLShiftC_16u_I_Ctx**(int nValue, *Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal left shift with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be used to left shift each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLShiftC_16s_I_Ctx**(int nValue, *Npp16s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal left shift with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be used to left shift each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLShiftC_32u_I_Ctx**(int nValue, *Npp32u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned signed integer in place signal left shift with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be used to left shift each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsLShiftC_32s_I_Ctx**(int nValue, *Npp32s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed signed integer in place signal left shift with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be used to left shift each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.17.2.10 Signal RShiftC

1.17.2.10.1 RShiftC

Right shifts the bits of each sample of a signal by a constant amount.

Functions

NppStatus **nppsRShiftC_8u_Ctx**(const *Npp8u* *pSrc, int nValue, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char signal right shift with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be used to right shift each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsRShiftC_16u_Ctx**(const *Npp16u* *pSrc, int nValue, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short signal right shift with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be used to right shift each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsRShiftC_16s_Ctx**(const *Npp16s* *pSrc, int nValue, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short signal right shift with constant.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nValue** – Constant value to be used to right shift each vector element
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsRShiftC_32u_Ctx**(const *Npp32u* *pSrc, int nValue, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer signal right shift with constant.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be used to right shift each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsRShiftC_32s_Ctx**(const *Npp32s* *pSrc, int nValue, *Npp32s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed integer signal right shift with constant.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nValue** – Constant value to be used to right shift each vector element
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsRShiftC_8u_I_Ctx**(int nValue, *Npp8u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char in place signal right shift with constant.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nValue** – Constant value to be used to right shift each vector element
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsRShiftC_16u_I_Ctx**(int nValue, *Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short in place signal right shift with constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be used to right shift each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsRShiftC_16s_I_Ctx**(int nValue, *Npp16s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short in place signal right shift with constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be used to right shift each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsRShiftC_32u_I_Ctx**(int nValue, *Npp32u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned signed integer in place signal right shift with constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be used to right shift each vector element
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsRShiftC_32s_I_Ctx**(int nValue, *Npp32s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed signed integer in place signal right shift with constant.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nValue** – Constant value to be used to right shift each vector element

- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.18. Signal Conversion Functions

Functions that provide conversion and threshold operations.

1.18.1. Signal Convert

1.18.1.1 Convert

The set of conversion operations available in the library

Convert

Routines for converting the sample-data type of signals.

NppStatus **nppsConvert_8s16s_Ctx**(const *Npp8s* *pSrc, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit signed byte to 16-bit signed short conversion

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_8s32f_Ctx**(const *Npp8s* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit signed byte to 32-bit floating point number conversion

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_8u32f_Ctx**(const *Npp8u* *pSrc, *Npp32f* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

8-bit unsigned byte to 32-bit floating point number conversion

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_16s8s_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp8s* *pDst, *Npp32u* nLength,
NppRoundMode eRoundMode, int nScaleFactor,
NppStreamContext nppStreamCtx)

16-bit signed short to 8-bit signed byte conversion

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **eRoundMode** – *Rounding Mode Parameter.*
- ▶ **nScaleFactor** – *nScaleFactor*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_16s32s_Ctx**(const *Npp16s* *pSrc, *Npp32s* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

16-bit signed short to 32-bit signed integer conversion

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_16s32f_Ctx**(const *Npp16s* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short to 32-bit floating point number conversion

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_16u32f_Ctx**(const *Npp16u* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short to 32-bit floating point number conversion

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_32s16s_Ctx**(const *Npp32s* *pSrc, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed integer to 16-bit signed short conversion

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_32s32f_Ctx**(const *Npp32s* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed integer to 32-bit floating point number conversion

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*

- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_32s64f_Ctx**(const *Npp32s* *pSrc, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed integer to 64-bit floating point number conversion

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_32f64f_Ctx**(const *Npp32f* *pSrc, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point number to 64-bit floating point number conversion

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_64s64f_Ctx**(const *Npp64s* *pSrc, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit signed integer to 64-bit floating point number conversion

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_64f32f_Ctx**(const *Npp64f* *pSrc, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point number to 32-bit floating point number conversion

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_16s32f_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp32f* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short to 32-bit floating point number conversion with scaling

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_16s64f_Sfs_Ctx**(const *Npp16s* *pSrc, *Npp64f* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

16-bit signed short to 64-bit floating point number conversion with scaling

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_32s16s_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp16s* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer to 16-bit signed short conversion with scaling

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*

- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_32s32f_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp32f* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer to 32-bit floating point number conversion with scaling

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_32s64f_Sfs_Ctx**(const *Npp32s* *pSrc, *Npp64f* *pDst, size_t nLength, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer to 64-bit floating point number conversion with scaling

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_32f8s_Sfs_Ctx**(const *Npp32f* *pSrc, *Npp8s* *pDst, size_t nLength, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit signed integer to 8-bit signed byte conversion with scaling

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **eRoundMode** – *Rounding Mode Parameter.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsConvert_32f8u_Sfs_Ctx**(const *Npp32f* *pSrc, *Npp8u* *pDst, size_t nLength,
NppRoundMode eRoundMode, int nScaleFactor,
NppStreamContext nppStreamCtx)

32-bit floating point number to 8-bit unsigned byte conversion with scaling

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **eRoundMode** – *Rounding Mode Parameter.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsConvert_32f16s_Sfs_Ctx**(const *Npp32f* *pSrc, *Npp16s* *pDst, size_t nLength,
NppRoundMode eRoundMode, int nScaleFactor,
NppStreamContext nppStreamCtx)

32-bit floating point number to 16-bit signed short conversion with scaling

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **eRoundMode** – *Rounding Mode Parameter.*
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsConvert_32f16u_Sfs_Ctx**(const *Npp32f* *pSrc, *Npp16u* *pDst, size_t nLength,
NppRoundMode eRoundMode, int nScaleFactor,
NppStreamContext nppStreamCtx)

32-bit floating point number to 16-bit unsigned short conversion with scaling

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **eRoundMode** – *Rounding Mode Parameter.*

- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_32f32s_Sfs_Ctx**(const *Npp32f* *pSrc, *Npp32s* *pDst, size_t nLength, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

32-bit floating point number to 32-bit signed integer conversion with scaling

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **eRoundMode** – *Rounding Mode Parameter.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_64s32s_Sfs_Ctx**(const *Npp64s* *pSrc, *Npp32s* *pDst, size_t nLength, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit signed integer to 32-bit signed integer conversion with scaling

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **eRoundMode** – *Rounding Mode Parameter.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_64f16s_Sfs_Ctx**(const *Npp64f* *pSrc, *Npp16s* *pDst, size_t nLength, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit floating point number to 16-bit signed short conversion with scaling

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*

- **nLength** – *Signal Length.*
- **eRoundMode** – *Rounding Mode Parameter.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_64f32s_Sfs_Ctx**(const *Npp64f* *pSrc, *Npp32s* *pDst, size_t nLength, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit floating point number to 32-bit signed integer conversion with scaling

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **eRoundMode** – *Rounding Mode Parameter.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsConvert_64f64s_Sfs_Ctx**(const *Npp64f* *pSrc, *Npp64s* *pDst, size_t nLength, *NppRoundMode* eRoundMode, int nScaleFactor, *NppStreamContext* nppStreamCtx)

64-bit floating point number to 64-bit signed integer conversion with scaling

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **eRoundMode** – *Rounding Mode Parameter.*
- **nScaleFactor** – *Integer Result Scaling.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.18.2. Signal Threshold

1.18.2.1 Threshold

The set of threshold operations available in the library.

Threshold Functions

Performs the threshold operation on the samples of a signal by limiting the sample values by a specified constant value.

NppStatus **nppsThreshold_16s_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, *Npp16s* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

16-bit signed short signal threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nRelOp** – *NppCmpOp* type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_16s_I_Ctx**(*Npp16s* *pSrcDst, size_t nLength, *Npp16s* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

16-bit in place signed short signal threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nRelOp** – *NppCmpOp* type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_16sc_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pDst, size_t nLength, *Npp16s* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

16-bit signed short complex number signal threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nRelOp** – NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_16sc_I_Ctx**(*Npp16sc* *pSrcDst, size_t nLength, *Npp16s* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

16-bit in place signed short complex number signal threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nRelOp** – NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *Npp32f* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

32-bit floating point signal threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value to be used to limit each signal sample
- ▶ **nRelOp** – NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *Npp32f* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

32-bit in place floating point signal threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nRelOp** – *NppCmpOp* type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pDst, size_t nLength, *Npp32f* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number signal threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nRelOp** – *NppCmpOp* type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_32fc_I_Ctx**(*Npp32fc* *pSrcDst, size_t nLength, *Npp32f* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

32-bit in place floating point complex number signal threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nRelOp** – *NppCmpOp* type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength, *Npp64f* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

64-bit floating point signal threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nRelOp** – *NppCmpOp* type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *Npp64f* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

64-bit in place floating point signal threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nRelOp** – *NppCmpOp* type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pDst, size_t nLength, *Npp64f* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number signal threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

- ▶ **nRelOp** – NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_64fc_I_Ctx**(*Npp64fc* *pSrcDst, size_t nLength, *Npp64f* nLevel, *NppCmpOp* nRelOp, *NppStreamContext* nppStreamCtx)

64-bit in place floating point complex number signal threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nRelOp** – NppCmpOp type of thresholding operation (NPP_CMP_LESS or NPP_CMP_GREATER only).
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_16s_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, *Npp16s* nLevel, *NppStreamContext* nppStreamCtx)

16-bit signed short signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_16s_I_Ctx**(*Npp16s* *pSrcDst, size_t nLength, *Npp16s* nLevel, *NppStreamContext* nppStreamCtx)

16-bit in place signed short signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_16sc_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pDst, size_t nLength, *Npp16s* nLevel, *NppStreamContext* nppStreamCtx)

16-bit signed short complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_16sc_I_Ctx**(*Npp16sc* *pSrcDst, size_t nLength, *Npp16s* nLevel, *NppStreamContext* nppStreamCtx)

16-bit in place signed short complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *Npp32f* nLevel, *NppStreamContext* nppStreamCtx)

32-bit floating point signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *Npp32f* nLevel, *NppStreamContext* nppStreamCtx)

32-bit in place floating point signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pDst, size_t nLength, *Npp32f* nLevel, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_32fc_I_Ctx**(*Npp32fc* *pSrcDst, size_t nLength, *Npp32f* nLevel, *NppStreamContext* nppStreamCtx)

32-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength, *Npp64f* nLevel, *NppStreamContext* nppStreamCtx)

64-bit floating point signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *Npp64f* nLevel, *NppStreamContext* nppStreamCtx)

64-bit in place floating point signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pDst, size_t nLength, *Npp64f* nLevel, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LT_64fc_I_Ctx**(*Npp64fc* *pSrcDst, size_t nLength, *Npp64f* nLevel, *NppStreamContext* nppStreamCtx)

64-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*

- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GT_16s_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, *Npp16s* nLevel, *NppStreamContext* nppStreamCtx)

16-bit signed short signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GT_16s_I_Ctx**(*Npp16s* *pSrcDst, size_t nLength, *Npp16s* nLevel, *NppStreamContext* nppStreamCtx)

16-bit in place signed short signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GT_16sc_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pDst, size_t nLength, *Npp16s* nLevel, *NppStreamContext* nppStreamCtx)

16-bit signed short complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsThreshold_GT_16sc_I_Ctx**(*Npp16sc* *pSrcDst, size_t nLength, *Npp16s* nLevel, *NppStreamContext* nppStreamCtx)

16-bit in place signed short complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsThreshold_GT_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *Npp32f* nLevel, *NppStreamContext* nppStreamCtx)

32-bit floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsThreshold_GT_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *Npp32f* nLevel, *NppStreamContext* nppStreamCtx)

32-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsThreshold_GT_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pDst, size_t nLength, *Npp32f* nLevel, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GT_32fc_I_Ctx**(*Npp32fc* *pSrcDst, size_t nLength, *Npp32f* nLevel, *NppStreamContext* nppStreamCtx)

32-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GT_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength, *Npp64f* nLevel, *NppStreamContext* nppStreamCtx)

64-bit floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GT_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *Npp64f* nLevel, *NppStreamContext* nppStreamCtx)

64-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*

- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GT_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pDst, size_t nLength, *Npp64f* nLevel, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GT_64fc_I_Ctx**(*Npp64fc* *pSrcDst, size_t nLength, *Npp64f* nLevel, *NppStreamContext* nppStreamCtx)

64-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_16s_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, *Npp16s* nLevel, *Npp16s* nValue, *NppStreamContext* nppStreamCtx)

16-bit signed short signal NPP_CMP_LESS threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample

- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_16s_I_Ctx**(*Npp16s* *pSrcDst, size_t nLength, *Npp16s* nLevel, *Npp16s* nValue, *NppStreamContext* nppStreamCtx)

16-bit in place signed short signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrcDst** – *In-Place Signal Pointer*.
- **nLength** – *Signal Length*.
- **nLevel** – Constant threshold value to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_16sc_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pDst, size_t nLength, *Npp16s* nLevel, *Npp16sc* nValue, *NppStreamContext* nppStreamCtx)

16-bit signed short complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrc** – *Source Signal Pointer*.
- **pDst** – *Destination Signal Pointer*.
- **nLength** – *Signal Length*.
- **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_16sc_I_Ctx**(*Npp16sc* *pSrcDst, size_t nLength, *Npp16s* nLevel, *Npp16sc* nValue, *NppStreamContext* nppStreamCtx)

16-bit in place signed short complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrcDst** – *In-Place Signal Pointer*.
- **nLength** – *Signal Length*.

- **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *Npp32f* nLevel, *Npp32f* nValue, *NppStreamContext* nppStreamCtx)

32-bit floating point signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nLevel1** – Constant threshold value to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *Npp32f* nLevel, *Npp32f* nValue, *NppStreamContext* nppStreamCtx)

32-bit in place floating point signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nLevel1** – Constant threshold value to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pDst, size_t nLength, *Npp32f* nLevel, *Npp32fc* nValue, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*

- **nLength** – *Signal Length*.
- **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_32fc_I_Ctx**(*Npp32fc* *pSrcDst, size_t nLength, *Npp32f* nLevel, *Npp32fc* nValue, *NppStreamContext* nppStreamCtx)

32-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrcDst** – *In-Place Signal Pointer*.
- **nLength** – *Signal Length*.
- **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength, *Npp64f* nLevel, *Npp64f* nValue, *NppStreamContext* nppStreamCtx)

64-bit floating point signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrc** – *Source Signal Pointer*.
- **pDst** – *Destination Signal Pointer*.
- **nLength** – *Signal Length*.
- **nLevel1** – Constant threshold value to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *Npp64f* nLevel, *Npp64f* nValue, *NppStreamContext* nppStreamCtx)

64-bit in place floating point signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nLevel1** – Constant threshold value to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pDst, size_t nLength, *Npp64f* nLevel, *Npp64fc* nValue, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_LTVa1_64fc_I_Ctx**(*Npp64fc* *pSrcDst, size_t nLength, *Npp64f* nLevel, *Npp64fc* nValue, *NppStreamContext* nppStreamCtx)

64-bit in place floating point complex number signal NPP_CMP_LESS threshold with constant level.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVa1_16s_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, *Npp16s* nLevel, *Npp16s* nValue, *NppStreamContext* nppStreamCtx)

16-bit signed short signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nValue** – Constant value to replace source value when threshold test is true.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVal_16s_I_Ctx**(*Npp16s* *pSrcDst, size_t nLength, *Npp16s* nLevel, *Npp16s* nValue, *NppStreamContext* nppStreamCtx)

16-bit in place signed short signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value to be used to limit each signal sample
- ▶ **nValue** – Constant value to replace source value when threshold test is true.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVal_16sc_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pDst, size_t nLength, *Npp16s* nLevel, *Npp16sc* nValue, *NppStreamContext* nppStreamCtx)

16-bit signed short complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nValue** – Constant value to replace source value when threshold test is true.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVa1_16sc_I_Ctx**(*Npp16sc* *pSrcDst, size_t nLength, *Npp16s* nLevel, *Npp16sc* nValue, *NppStreamContext* nppStreamCtx)

16-bit in place signed short complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nValue** – Constant value to replace source value when threshold test is true.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVa1_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength, *Npp32f* nLevel, *Npp32f* nValue, *NppStreamContext* nppStreamCtx)

32-bit floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value to be used to limit each signal sample
- ▶ **nValue** – Constant value to replace source value when threshold test is true.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVa1_32f_I_Ctx**(*Npp32f* *pSrcDst, size_t nLength, *Npp32f* nLevel, *Npp32f* nValue, *NppStreamContext* nppStreamCtx)

32-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value to be used to limit each signal sample
- ▶ **nValue** – Constant value to replace source value when threshold test is true.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVa1_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pDst, size_t nLength, *Npp32f* nLevel, *Npp32fc* nValue, *NppStreamContext* nppStreamCtx)

32-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nValue** – Constant value to replace source value when threshold test is true.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVa1_32fc_I_Ctx**(*Npp32fc* *pSrcDst, size_t nLength, *Npp32f* nLevel, *Npp32fc* nValue, *NppStreamContext* nppStreamCtx)

32-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nValue** – Constant value to replace source value when threshold test is true.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVa1_64f_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pDst, size_t nLength, *Npp64f* nLevel, *Npp64f* nValue, *NppStreamContext* nppStreamCtx)

64-bit floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*

- **nLevel1** – Constant threshold value to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVal1_64f_I_Ctx**(*Npp64f* *pSrcDst, size_t nLength, *Npp64f* nLevel, *Npp64f* nValue, *NppStreamContext* nppStreamCtx)

64-bit in place floating point signal NPP_CMP_GREATER threshold with constant level.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*
- **nLength** – *Signal Length.*
- **nLevel1** – Constant threshold value to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVal1_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pDst, size_t nLength, *Npp64f* nLevel, *Npp64fc* nValue, *NppStreamContext* nppStreamCtx)

64-bit floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nLevel1** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- **nValue** – Constant value to replace source value when threshold test is true.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsThreshold_GTVal1_64fc_I_Ctx**(*Npp64fc* *pSrcDst, size_t nLength, *Npp64f* nLevel, *Npp64fc* nValue, *NppStreamContext* nppStreamCtx)

64-bit in place floating point complex number signal NPP_CMP_GREATER threshold with constant level.

Parameters

- **pSrcDst** – *In-Place Signal Pointer.*

- ▶ **nLength** – *Signal Length*.
- ▶ **nLevel** – Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample
- ▶ **nValue** – Constant value to replace source value when threshold test is true.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.19. Signal Filtering Functions

Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.

1.19.1. Integral

Compute the indefinite integral of a given signal. The i-th element is computed to be

$$s'_i = \sum_0^i s_j$$

Functions

NppStatus **nppsIntegralGetBufferSize_32s**(size_t nLength, size_t *hpBufferSize)

Device scratch buffer size (in bytes) for 32s nppsIntegral.

This primitive provides the correct buffer size for nppsIntegral_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.

NppStatus **nppsIntegral_32s_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pDst, size_t nLength, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Compute cumulative sum of 32-bit signed integer signal.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **pDst** – Pointer to the output result.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

1.20. Signal Initialization Functions

Functions that provide functionality of initialization signal like: set, zero or copy other signal.

1.20.1. Signal Set

1.20.1.1 Set

The set of set initialization operations available in the library.

Set

Set methods for 1D vectors of various types.

The copy methods operate on vector data given as a pointer to the underlying data-type (e.g. 8-bit vectors would be passed as pointers to Npp8u type) and length of the vectors, i.e. the number of items.

NppStatus **nppsSet_8u_Ctx**(*Npp8u* nValue, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_8s_Ctx**(*Npp8s* nValue, *Npp8s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit signed char, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_16u_Ctx**(*Npp16u* nValue, *Npp16u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned integer, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_16s_Ctx**(*Npp16s* nValue, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed integer, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_16sc_Ctx**(*Npp16sc* nValue, *Npp16sc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit integer complex, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_32u_Ctx**(*Npp32u* nValue, *Npp32u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit unsigned integer, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_32s_Ctx**(*Npp32s* nValue, *Npp32s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed integer, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_32sc_Ctx**(*Npp32sc* nValue, *Npp32sc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit integer complex, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_32f_Ctx**(*Npp32f* nValue, *Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit float, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_32fc_Ctx**(*Npp32fc* nValue, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit float complex, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_64s_Ctx**(*Npp64s* nValue, *Npp64s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit long long integer, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_64sc_Ctx**(*Npp64sc* nValue, *Npp64sc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit long long integer complex, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_64f_Ctx**(*Npp64f* nValue, *Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit double, vector set method.

Parameters

- ▶ **nValue** – Value used to initialize the vector pDst.
- ▶ **pDst** – *Destination Signal Pointer*.

- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSet_64fc_Ctx**(*Npp64fc* nValue, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit double complex, vector set method.

Parameters

- **nValue** – Value used to initialize the vector pDst.
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.20.2. Signal Zero

1.20.2.1 Zero

The set of zero initialization operations available in the library.

Zero

Set signals to zero.

NppStatus **nppsZero_8u_Ctx**(*Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char, vector zero method.

Parameters

- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsZero_16s_Ctx**(*Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit integer, vector zero method.

Parameters

- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsZero_16sc_Ctx**(*Npp16sc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit integer complex, vector zero method.

Parameters

- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsZero_32s_Ctx**(*Npp32s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit integer, vector zero method.

Parameters

- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsZero_32sc_Ctx**(*Npp32sc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit integer complex, vector zero method.

Parameters

- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsZero_32f_Ctx**(*Npp32f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit float, vector zero method.

Parameters

- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsZero_32fc_Ctx**(*Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit float complex, vector zero method.

Parameters

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsZero_64s_Ctx**(*Npp64s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit long long integer, vector zero method.

Parameters

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsZero_64sc_Ctx**(*Npp64sc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit long long integer complex, vector zero method.

Parameters

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsZero_64f_Ctx**(*Npp64f* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit double, vector zero method.

Parameters

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsZero_64fc_Ctx**(*Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit double complex, vector zero method.

Parameters

- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.20.3. Signal Copy

1.20.3.1 Copy

The set of copy initialization operations available in the library.

Copy

Copy methods for various type signals.

Copy methods operate on signal data given as a pointer to the underlying data-type (e.g. 8-bit vectors would be passed as pointers to *Npp8u* type) and length of the vectors, i.e. the number of items.

NppStatus **nppsCopy_8u_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit unsigned char, vector copy method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCopy_16s_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short, vector copy method.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCopy_32s_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

32-bit signed integer, vector copy method.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCopy_32f_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

32-bit float, vector copy method.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCopy_64s_Ctx**(const *Npp64s* *pSrc, *Npp64s* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

64-bit signed integer, vector copy method.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **pDst** – *Destination Signal Pointer.*
- **nLength** – *Signal Length.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCopy_16sc_Ctx**(const *Npp16sc* *pSrc, *Npp16sc* *pDst, size_t nLength,
NppStreamContext nppStreamCtx)

16-bit complex short, vector copy method.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCopy_32sc_Ctx**(const *Npp32sc* *pSrc, *Npp32sc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex signed integer, vector copy method.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCopy_32fc_Ctx**(const *Npp32fc* *pSrc, *Npp32fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit complex float, vector copy method.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCopy_64sc_Ctx**(const *Npp64sc* *pSrc, *Npp64sc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex signed integer, vector copy method.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsCopy_64fc_Ctx**(const *Npp64fc* *pSrc, *Npp64fc* *pDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit complex double, vector copy method.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pDst** – *Destination Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21. Signal Statistical Functions

Functions that provide global signal statistics like: sum, mean, standard deviation, min, max, etc.

1.21.1. Signal Min Every Or Max Every

1.21.1.1 MinEvery And MaxEvery Functions

Performs the min or max operation on the samples of a signal.

Functions

NppStatus **nppsMinEvery_8u_I_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit in place min value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinEvery_16u_I_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short integer in place min value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinEvery_16s_I_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short integer in place min value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinEvery_32s_I_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed integer in place min value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinEvery_32f_I_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place min value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMinEvery_64f_I_Ctx**(const *Npp64f* *pSrc, *Npp64f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

64-bit floating point in place min value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMaxEvery_8u_I_Ctx**(const *Npp8u* *pSrc, *Npp8u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

8-bit in place max value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMaxEvery_16u_I_Ctx**(const *Npp16u* *pSrc, *Npp16u* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit unsigned short integer in place max value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMaxEvery_16s_I_Ctx**(const *Npp16s* *pSrc, *Npp16s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

16-bit signed short integer in place max value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*

- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaxEvery_32s_I_Ctx**(const *Npp32s* *pSrc, *Npp32s* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit signed integer in place max value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaxEvery_32f_I_Ctx**(const *Npp32f* *pSrc, *Npp32f* *pSrcDst, size_t nLength, *NppStreamContext* nppStreamCtx)

32-bit floating point in place max value for each pair of elements.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **pSrcDst** – *In-Place Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.2. Signal Sum

signal_min_every_or_max_every

1.21.2.1 Sum

Performs the sum operation on the samples of a signal.

Functions

NppStatus **nppsSumGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsSum_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsSumGetBufferSize_32fc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsSum_32fc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsSumGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsSum_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsSumGetBufferSize_64fc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsSum_64fc.

Parameters

- ▶ **nLength** – *Signal Length*.

- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsSumGetBufferSize_16s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsSum_16s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsSumGetBufferSize_16sc_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsSum_16sc_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsSumGetBufferSize_16sc32sc_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsSum_16sc32sc_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsSumGetBufferSize_32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsSum_32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsSumGetBufferSize_16s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsSum_16s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsSum_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pSum, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector sum method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pSum** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsSumGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSum_32fc_Ctx**(const *Npp32fc* *pSrc, size_t nLength, *Npp32fc* *pSum, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float complex vector sum method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pSum** – Device memory pointer to the output result.

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsSumGetBufferSize_32fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSum_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pSum, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit double vector sum method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pSum** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsSumGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSum_64fc_Ctx**(const *Npp64fc* *pSrc, size_t nLength, *Npp64fc* *pSum, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit double complex vector sum method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pSum** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsSumGetBufferSize_64fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSum_16s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pSum, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit short vector sum with integer scaling method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.

- ▶ **pSum** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsSumGetBufferSize_16s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSum_32s_Sfs_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pSum, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector sum with integer scaling method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pSum** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsSumGetBufferSize_32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSum_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc, size_t nLength, *Npp16sc* *pSum, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit short complex vector sum with integer scaling method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pSum** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsSumGetBufferSize_16sc_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSum_16sc32sc_Sfs_Ctx**(const *Npp16sc* *pSrc, size_t nLength, *Npp32sc* *pSum, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit short complex vector sum (32bit int complex) with integer scaling method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pSum** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsSumGetBufferSize_16sc32sc_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsSum_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32s* *pSum, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit integer vector sum (32bit) with integer scaling method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pSum** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsSumGetBufferSize_16s32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.3. Signal Maximum

1.21.3.1 Maximum

Performs the maximum operation on the samples of a signal.

Functions

NppStatus **nppsMaxGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMax_16s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaxGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMax_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaxGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMax_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaxGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMax_64f.

Parameters

- ▶ **nLength** – *Signal Length*.

- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMax_16s_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit integer vector max method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMax** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaxGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMax_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector max method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMax** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaxGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMax_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector max method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMax** – Device memory pointer to the output result.

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaxGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMax_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float vector max method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMax** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaxGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaxIdxGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMaxIdx_16s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaxIdxGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMaxIdx_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaxIdxGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMaxIdx_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaxIdxGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMaxIdx_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaxIdx_16s_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMax, int *pIdx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit integer vector max index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMax** – Device memory pointer to the output result.
- ▶ **pIdx** – Device memory pointer to the index value of the first maximum element.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaxIdxGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaxIdx_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMax, int *pIdx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector max index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMax** – Device memory pointer to the output result.
- ▶ **pIndx** – Device memory pointer to the index value of the first maximum element.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaxIndxGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaxIndx_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pMax, int *pIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector max index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMax** – Device memory pointer to the output result.
- ▶ **pIndx** – Device memory pointer to the index value of the first maximum element.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaxIndxGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaxIndx_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pMax, int *pIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float vector max index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMax** – Device memory pointer to the output result.
- ▶ **pIndx** – Device memory pointer to the index value of the first maximum element.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaxIndxGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaxAbsGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMaxAbs_16s_Ctx.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaxAbsGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMaxAbs_32s.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaxAbs_16s_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMaxAbs, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit integer vector max absolute method

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nLength** – *Signal Length.*
- **pMaxAbs** – Device memory pointer to the output result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMaxAbsGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaxAbs_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMaxAbs, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector max absolute method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMaxAbs** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMaxAbsGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaxAbsIndxGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMaxAbsIndx_16s.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaxAbsIndxGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMaxAbsIndx_32s.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaxAbsIndx_16s_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMaxAbs, int *pIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit integer vector max absolute index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*

- ▶ **pMaxAbs** – Device memory pointer to the output result.
- ▶ **pIndx** – Device memory pointer to the index value of the first maximum element.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaxAbsIndxGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaxAbsIndx_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMaxAbs, int *pIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector max absolute index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMaxAbs** – Device memory pointer to the output result.
- ▶ **pIndx** – Device memory pointer to the index value of the first maximum element.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaxAbsIndxGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.4. Signal Minimum

1.21.4.1 Minimum

Performs the minimum operation on the samples of a signal.

Functions

NppStatus **nppsMinGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMin_16s.

Parameters

- ▶ **nLength** – *Signal Length*.

- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMin_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMin_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMin_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMin_16s_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMin, *Npp8u*
 *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit integer vector min method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMin** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMinGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMin_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMin, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector min method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMin** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMinGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMin_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pMin, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector min method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMin** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMinGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMin_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pMin, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit integer vector min method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinIndxGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMinIndx_16s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinIndxGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMinIndx_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinIndxGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMinIndx_32f.

Parameters

- ▶ **nLength** – *Signal Length*.

- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinIdxGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMinIdx_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinIdx_16s_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMin, int *pIdx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit integer vector min index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the output result.
- ▶ **pIdx** – Device memory pointer to the index value of the first minimum element.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinIdxGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMinIdx_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMin, int *pIdx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector min index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the output result.
- ▶ **pIdx** – Device memory pointer to the index value of the first minimum element.

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinIndxGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinIndx_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pMin, int *pIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector min index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the output result.
- ▶ **pIndx** – Device memory pointer to the index value of the first minimum element.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinIndxGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinIndx_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pMin, int *pIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float vector min index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the output result.
- ▶ **pIndx** – Device memory pointer to the index value of the first minimum element.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinIndxGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinAbsGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMinAbs_16s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinAbsGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMinAbs_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinAbs_16s_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMinAbs, *Npp8u*
*pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit integer vector min absolute method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMinAbs** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinAbsGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMinAbs_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMinAbs, *Npp8u*
*pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector min absolute method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMinAbs** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinAbsGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinAbsIndxGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMinAbsIndx_16s.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMinAbsIndxGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMinAbsIndx_32s.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer.*
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMinAbsIndx_16s_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMinAbs, int *pIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit integer vector min absolute index method

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nLength** – *Signal Length.*
- **pMinAbs** – Device memory pointer to the output result.
- **pIndx** – Device memory pointer to the index value of the first minimum element.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMinAbsIndxGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinAbsIndx_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMinAbs, int *pIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector min absolute index method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMinAbs** – Device memory pointer to the output result.
- ▶ **pIndx** – Device memory pointer to the index value of the first minimum element.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinAbsIndxGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.5. Signal Mean

1.21.5.1 Mean

Performs the mean operation on the samples of a signal.

Functions

NppStatus **nppsMeanGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMean_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMeanGetBufferSize_32fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMean_32fc.

Parameters

- ▶ **nLength** – *Signal Length*.

- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMeanGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMean_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMeanGetBufferSize_64fc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMean_64fc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMeanGetBufferSize_16s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMean_16s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMeanGetBufferSize_32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMean_32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMeanGetBufferSize_16sc_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMean_16sc_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMean_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pMean, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector mean method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMean** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMeanGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMean_32fc_Ctx**(const *Npp32fc* *pSrc, size_t nLength, *Npp32fc* *pMean, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float complex vector mean method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMean** – Device memory pointer to the output result.

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMeanGetBufferSize_32fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMean_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pMean, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit double vector mean method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMean** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMeanGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMean_64fc_Ctx**(const *Npp64fc* *pSrc, size_t nLength, *Npp64fc* *pMean, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit double complex vector mean method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMean** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMeanGetBufferSize_64fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMean_16s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMean, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit short vector mean with integer scaling method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.

- ▶ **pMean** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMeanGetBufferSize_16s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMean_32s_Sfs_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMean, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit integer vector mean with integer scaling method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMean** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMeanGetBufferSize_32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMean_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc, size_t nLength, *Npp16sc* *pMean, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit short complex vector mean with integer scaling method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMean** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMeanGetBufferSize_16sc_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.6. Signal StdDev

1.21.6.1 Standard Deviation

Calculates the standard deviation for the samples of a signal.

Functions

NppStatus **nppsStdDevGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsStdDev_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsStdDevGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsStdDev_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsStdDevGetBufferSize_16s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsStdDev_16s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsStdDevGetBufferSize_16s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsStdDev_16s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsStdDev_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pStdDev, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector standard deviation method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pStdDev** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsStdDevGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsStdDev_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pStdDev, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float vector standard deviation method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pStdDev** – Device memory pointer to the output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsStdDevGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsStdDev_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32s* *pStdDev, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit float vector standard deviation method (return value is 32-bit)

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pStdDev** – Device memory pointer to the output result.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsStdDevGetBufferSize_16s32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsStdDev_16s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pStdDev, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit float vector standard deviation method (return value is also 16-bit)

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pStdDev** – Device memory pointer to the output result.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsStdDevGetBufferSize_16s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.7. Signal Mean And StdDev

1.21.7.1 Mean And Standard Deviation

Performs the mean and calculates the standard deviation for the samples of a signal.

Functions

NppStatus **nppsMeanStdDevGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMeanStdDev_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMeanStdDevGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMeanStdDev_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMeanStdDevGetBufferSize_16s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMeanStdDev_16s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer. Scratch Buffer and Host Pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMeanStdDevGetBufferSize_16s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device scratch buffer size (in bytes) for nppsMeanStdDev_16s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.

- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*. *Scratch Buffer and Host Pointer*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMeanStdDev_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pMean, *Npp32f* *pStdDev, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector mean and standard deviation method

Parameters

- **pSrc** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pMean** – Device memory pointer to the output mean value.
- **pStdDev** – Device memory pointer to the output standard deviation value.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMeanStdDevGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMeanStdDev_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pMean, *Npp64f* *pStdDev, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float vector mean and standard deviation method

Parameters

- **pSrc** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pMean** – Device memory pointer to the output mean value.
- **pStdDev** – Device memory pointer to the output standard deviation value.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMeanStdDevGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMeanStdDev_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32s* *pMean, *Npp32s* *pStdDev, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit float vector mean and standard deviation method (return values are 32-bit)

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMean** – Device memory pointer to the output mean value.
- ▶ **pStdDev** – Device memory pointer to the output standard deviation value.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMeanStdDevGetBufferSize_16s32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMeanStdDev_16s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMean, *Npp16s* *pStdDev, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit float vector mean and standard deviation method (return values are also 16-bit)

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMean** – Device memory pointer to the output mean value.
- ▶ **pStdDev** – Device memory pointer to the output standard deviation value.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMeanStdDevGetBufferSize_16s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.8. Signal MinMax

1.21.8.1 Minimum Maximum

Performs the maximum and the minimum operation on the samples of a signal.

Functions

NppStatus **nppsMinMaxGetBufferSize_8u_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMax_8u.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMax_16s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxGetBufferSize_16u_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMax_16u.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMax_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxGetBufferSize_32u_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMax_32u.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMax_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMax_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMax_8u_Ctx**(const *Npp8u* *pSrc, size_t nLength, *Npp8u* *pMin, *Npp8u* *pMax,
Npp8u *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

8-bit char vector min and max method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMax** – Device memory pointer to the max output result.

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxGetBufferSize_8u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinMax_16s_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMin, *Npp16s* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short vector min and max method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinMax_16u_Ctx**(const *Npp16u* *pSrc, size_t nLength, *Npp16u* *pMin, *Npp16u* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit unsigned short vector min and max method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxGetBufferSize_16u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinMax_32u_Ctx**(const *Npp32u* *pSrc, size_t nLength, *Npp32u* *pMin, *Npp32u* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit unsigned int vector min and max method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxGetBufferSize_32u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinMax_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMin, *Npp32s* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit signed int vector min and max method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinMax_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pMin, *Npp32f* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector min and max method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinMax_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pMin, *Npp64f* *pMax, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit double vector min and max method

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nLength** – *Signal Length.*
- **pMin** – Device memory pointer to the min output result.
- **pMax** – Device memory pointer to the max output result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinMaxIndxGetBufferSize_8u_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMaxIndx_8u.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxIndxGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMaxIndx_16s.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxIndxGetBufferSize_16u_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMaxIndx_16u.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxIndxGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMaxIndx_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxIndxGetBufferSize_32u_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMaxIndx_32u.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxIndxGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMaxIndx_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxIndxGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMinMaxIndx_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMinMaxIndx_8u_Ctx**(const *Npp8u* *pSrc, size_t nLength, *Npp8u* *pMin, int *pMinIndx, *Npp8u* *pMax, int *pMaxIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

8-bit char vector min and max with indices method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMinIndx** – Device memory pointer to the index of the first min value.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pMaxIndx** – Device memory pointer to the index of the first max value.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxIndxGetBufferSize_8u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinMaxIndx_16s_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp16s* *pMin, int *pMinIndx, *Npp16s* *pMax, int *pMaxIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short vector min and max with indices method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMinIndx** – Device memory pointer to the index of the first min value.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pMaxIndx** – Device memory pointer to the index of the first max value.

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxIdxGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinMaxIdx_16u_Ctx**(const *Npp16u* *pSrc, size_t nLength, *Npp16u* *pMin, int *pMinIdx, *Npp16u* *pMax, int *pMaxIdx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit unsigned short vector min and max with indices method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMinIdx** – Device memory pointer to the index of the first min value.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pMaxIdx** – Device memory pointer to the index of the first max value.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxIdxGetBufferSize_16u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMinMaxIdx_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, *Npp32s* *pMin, int *pMinIdx, *Npp32s* *pMax, int *pMaxIdx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit signed short vector min and max with indices method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMinIdx** – Device memory pointer to the index of the first min value.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pMaxIdx** – Device memory pointer to the index of the first max value.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxIdxGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMinMaxIndx_32u_Ctx**(const *Npp32u* *pSrc, size_t nLength, *Npp32u* *pMin, int *pMinIndx, *Npp32u* *pMax, int *pMaxIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit unsigned short vector min and max with indices method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMinIndx** – Device memory pointer to the index of the first min value.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pMaxIndx** – Device memory pointer to the index of the first max value.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxIndxGetBufferSize_32u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMinMaxIndx_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pMin, int *pMinIndx, *Npp32f* *pMax, int *pMaxIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector min and max with indices method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMinIndx** – Device memory pointer to the index of the first min value.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pMaxIndx** – Device memory pointer to the index of the first max value.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxIndxGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMinMaxIndx_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pMin, int *pMinIndx, *Npp64f* *pMax, int *pMaxIndx, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float vector min and max with indices method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pMin** – Device memory pointer to the min output result.
- ▶ **pMinIndx** – Device memory pointer to the index of the first min value.
- ▶ **pMax** – Device memory pointer to the max output result.
- ▶ **pMaxIndx** – Device memory pointer to the index of the first max value.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMinMaxIndxGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.9. Signal Norms

1.21.9.1 Signal Norm Inf

1.21.9.1.1 Infinity Norm

Performs the infinity norm on the samples of a signal.

Functions

NppStatus **nppsNormInfGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_Inf_32f.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNorm_Inf_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector C norm method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormInfGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormInfGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_Inf_64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNorm_Inf_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float vector C norm method

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormInfGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormInfGetBufferSize_16s32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_Inf_16s32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*

- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_Inf_16s32f_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer vector C norm method, return value is 32-bit float.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormInfGetBufferSize_16s32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsNormInfGetBufferSize_32fc32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_Inf_32fc32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_Inf_32fc32f_Ctx**(const *Npp32fc* *pSrc, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float complex vector C norm method, return value is 32-bit float.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormInfGetBufferSize_32fc32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormInfGetBufferSize_64fc64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_Inf_64fc64f.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNorm_Inf_64fc64f_Ctx**(const *Npp64fc* *pSrc, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float complex vector C norm method, return value is 64-bit float.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormInfGetBufferSize_64fc64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormInfGetBufferSize_16s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_Inf_16s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNorm_Inf_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32s* *pNorm, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer vector C norm method, return value is 32-bit signed integer.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormInfGetBufferSize_16s32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.9.2 Signal Norm L1**1.21.9.2.1 L1 Norm**

Performs the L1 norm on the samples of a signal.

Functions

NppStatus **nppsNormL1GetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L1_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L1_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector L1 norm method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL1GetBufferSize_32f_Ctx* to determine the minimum number of bytes required.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL1GetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L1_64f_Ctx.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L1_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float vector L1 norm method

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nLength** – *Signal Length.*
- **pNorm** – Device memory pointer to the norm result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL1GetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL1GetBufferSize_16s32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L1_16s32f_Ctx.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L1_16s32f_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer vector L1 norm method, return value is 32-bit float.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the L1 norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL1GetBufferSize_16s32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL1GetBufferSize_32fc64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L1_32fc64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L1_32fc64f_Ctx**(const *Npp32fc* *pSrc, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float complex vector L1 norm method, return value is 64-bit float.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL1GetBufferSize_32fc64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL1GetBufferSize_64fc64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L1_64fc64f.

Parameters

- ▶ **nLength** – *Signal Length.*

- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L1_64fc64f_Ctx**(const *Npp64fc* *pSrc, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float complex vector L1 norm method, return value is 64-bit float.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL1GetBufferSize_64fc64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, *Length Related Error Codes*.

NppStatus **nppsNormL1GetBufferSize_16s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L1_16s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L1_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32s* *pNorm, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer vector L1 norm method, return value is 32-bit signed integer.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL1GetBufferSize_16s32s_Sfs_Ctx* to determine the minimum number of bytes required.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL1GetBufferSize_16s64s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L1_16s64s_Sfs_Ctx.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L1_16s64s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp64s* *pNorm,
int nScaleFactor, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

16-bit signed short integer vector L1 norm method, return value is 64-bit signed integer.

Parameters

- **pSrc** – *Source Signal Pointer.*
- **nLength** – *Signal Length.*
- **pNorm** – Device memory pointer to the norm result.
- **nScaleFactor** – *Integer Result Scaling.*
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL1GetBufferSize_16s64s_Sfs_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.9.3 Signal Norm L2

1.21.9.3.1 L2 Norm

Performs the L2 norm on the samples of a signal.

Functions

NppStatus **nppsNormL2GetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L2_32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L2_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float vector L2 norm method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL2GetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL2GetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L2_64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L2_64f_Ctx**(const *Npp64f* *pSrc, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float vector L2 norm method

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.

- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL2GetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL2GetBufferSize_16s32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L2_16s32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L2_16s32f_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer vector L2 norm method, return value is 32-bit float.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL2GetBufferSize_16s32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL2GetBufferSize_32fc64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L2_32fc64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L2_32fc64f_Ctx**(const *Npp32fc* *pSrc, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float complex vector L2 norm method, return value is 64-bit float.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsNormL2GetBufferSize_32fc64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL2GetBufferSize_64fc64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L2_64fc64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L2_64fc64f_Ctx**(const *Npp64fc* *pSrc, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float complex vector L2 norm method, return value is 64-bit float.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsNormL2GetBufferSize_64fc64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL2GetBufferSize_16s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L2_16s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L2_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32s* *pNorm, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer vector L2 norm method, return value is 32-bit signed integer.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormL2GetBufferSize_16s32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormL2SqrGetBufferSize_16s64s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNorm_L2Sqr_16s64s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNorm_L2Sqr_16s64s_Sfs_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp64s* *pNorm, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer vector L2 Square norm method, return value is 64-bit signed integer.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsNormL2SqrGetBufferSize_16s64s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.9.4 Signal Norm Inf NormDiff**1.21.9.4.1 Infinity Norm Diff**

Performs the infinity norm on the samples of two input signals' difference.

Functions

NppStatus **nppsNormDiffInfGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_Inf_32f.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_Inf_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float C norm method on two vectors' difference

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffInfGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffInfGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_Inf_64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_Inf_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float C norm method on two vectors' difference

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffInfGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffInfGetBufferSize_16s32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_Inf_16s32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_Inf_16s32f_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit float.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffInfGetBufferSize_16s32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffInfGetBufferSize_32fc32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_Inf_32fc32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_Inf_32fc32f_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float complex C norm method on two vectors' difference, return value is 32-bit float.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffInfGetBufferSize_32fc32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffInfGetBufferSize_64fc64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_Inf_64fc64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_Inf_64fc64f_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float complex C norm method on two vectors' difference, return value is 64-bit float.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffInfGetBufferSize_64fc64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffInfGetBufferSize_16s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_Inf_16s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_Inf_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp32s* *pNorm, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit signed integer.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffInfGetBufferSize_16s32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.9.5 Signal Norm L1 NormDiff

1.21.9.5.1 L1 Norm Diff

Performs the L1 norm on the samples of two input signals' difference.

Functions

NppStatus **nppsNormDiffL1GetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L1_32f.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L1_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float L1 norm method on two vectors' difference

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*

- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL1GetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffL1GetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L1_64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L1_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float L1 norm method on two vectors' difference

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL1GetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffL1GetBufferSize_16s32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L1_16s32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.

- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L1_16s32f_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit float.

Parameters

- **pSrc1** – *Source Signal Pointer*.
- **pSrc2** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pNorm** – Device memory pointer to the L1 norm result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL1GetBufferSize_16s32f_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsNormDiffL1GetBufferSize_32fc64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L1_32fc64f_Ctx.

Parameters

- **nLength** – *Signal Length*.
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L1_32fc64f_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.

Parameters

- **pSrc1** – *Source Signal Pointer*.
- **pSrc2** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pNorm** – Device memory pointer to the norm result.

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL1GetBufferSize_32fc64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffL1GetBufferSize_64fc64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L1_64fc64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L1_64fc64f_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL1GetBufferSize_64fc64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffL1GetBufferSize_16s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L1_16s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L1_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp32s* *pNorm, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit signed integer.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer..*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsNormDiffL1GetBufferSize_16s32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsNormDiffL1GetBufferSize_16s64s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L1_16s64s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L1_16s64s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp64s* *pNorm, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer L1 norm method on two vectors' difference, return value is 64-bit signed integer.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*

- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL1GetBufferSize_16s64s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.9.6 Signal Norm L2 NormDiff

1.21.9.6.1 L2 Norm Diff

Performs the L2 norm on the samples of two input signals' difference.

Functions

NppStatus **nppsNormDiffL2GetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L2_32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L2_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, size_t nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float L2 norm method on two vectors' difference

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL2GetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsNormDiffL2GetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L2_64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L2_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, size_t
 nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

64-bit float L2 norm method on two vectors' difference

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL2GetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsNormDiffL2GetBufferSize_16s32f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L2_16s32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L2_16s32f_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t
 nLength, *Npp32f* *pNorm, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit float.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL2GetBufferSize_16s32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffL2GetBufferSize_32fc64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L2_32fc64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L2_32fc64f_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL2GetBufferSize_32fc64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffL2GetBufferSize_64fc64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L2_64fc64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L2_64fc64f_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, size_t nLength, *Npp64f* *pNorm, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL2GetBufferSize_64fc64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffL2GetBufferSize_16s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L2_16s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L2_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp32s* *pNorm, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit signed integer.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL2GetBufferSize_16s32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs_Ctx**(size_t nLength, size_t
*hpBufferSize,
NppStreamContext
nppStreamCtx)

Device-buffer size (in bytes) for nppsNormDiff_L2Sqr_16s64s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsNormDiff_L2Sqr_16s64s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2,
size_t nLength, *Npp64s* *pNorm, int
nScaleFactor, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

16-bit signed short integer L2 Square norm method on two vectors' difference, return value is 64-bit signed integer.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pNorm** – Device memory pointer to the norm result.
- ▶ **nScaleFactor** – *Integer Result Scaling.*

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.10. Signal Dot Product

1.21.10.1 Dot Product

Performs the dot product operation on the samples of two input signals.

Functions

NppStatus **nppsDotProdGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, size_t nLength, *Npp32f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float dot product method, return value is 32-bit float.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_32fc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_32fc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_32fc_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, size_t nLength,
Npp32fc *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext*
nppStreamCtx)

32-bit float complex dot product method, return value is 32-bit float complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_32fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_32f32fc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_32f32fc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_32f32fc_Ctx**(const *Npp32f* *pSrc1, const *Npp32fc* *pSrc2, size_t
nLength, *Npp32fc* *pDp, *Npp8u* *pDeviceBuffer,
NppStreamContext nppStreamCtx)

32-bit float and 32-bit float complex dot product method, return value is 32-bit float complex.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer.*
- **nLength** – *Signal Length.*
- **pDp** – Device memory pointer to the dot product result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_32f32fc_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_32f64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_32f64f_Ctx.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_32f64f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, size_t nLength, *Npp64f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float dot product method, return value is 64-bit float.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer.*
- **nLength** – *Signal Length.*
- **pDp** – Device memory pointer to the dot product result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_32f64f_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_32fc64fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_32fc64fc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_32fc64fc_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, size_t nLength, *Npp64fc* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float complex dot product method, return value is 64-bit float complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_32fc64fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsDotProdGetBufferSize_32f32fc64fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_32f32fc64fc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_32f32fc64fc_Ctx**(const *Npp32f* *pSrc1, const *Npp32fc* *pSrc2, size_t nLength, *Npp64fc* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit float and 32-bit float complex dot product method, return value is 64-bit float complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.

- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_32f32fc64fc_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_64f_Ctx.

Parameters

- **nLength** – *Signal Length*.
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, size_t nLength, *Npp64f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float dot product method, return value is 64-bit float.

Parameters

- **pSrc1** – *Source Signal Pointer*.
- **pSrc2** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pDp** – Device memory pointer to the dot product result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_64fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_64fc_Ctx.

Parameters

- **nLength** – *Signal Length*.
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_64fc_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, size_t nLength, *Npp64fc* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float complex dot product method, return value is 64-bit float complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_64fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_64f64fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_64f64fc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_64f64fc_Ctx**(const *Npp64f* *pSrc1, const *Npp64fc* *pSrc2, size_t nLength, *Npp64fc* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit float and 64-bit float complex dot product method, return value is 64-bit float complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_64f64fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16s64s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16s64s_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16s64s_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp64s* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer dot product method, return value is 64-bit signed integer.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16s64s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16sc64sc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16sc64sc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16sc64sc_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp64sc* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer complex dot product method, return value is 64-bit signed integer complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16sc64sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16s16sc64sc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16s16sc64sc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16s16sc64sc_Ctx**(const *Npp16s* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp64sc* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer and 16-bit signed short integer short dot product method, return value is 64-bit signed integer complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16s16sc64sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16s32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16s32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16s32f_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp32f* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer dot product method, return value is 32-bit float.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16s32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16sc32fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16sc32fc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16sc32fc_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp32fc* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer complex dot product method, return value is 32-bit float complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16sc32fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16s16sc32fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16s16sc32fc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16s16sc32fc_Ctx**(const *Npp16s* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp32fc* *pDp, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit float complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16s16sc32fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16s_Sfs_Ctx.

Parameters

- **nLength** – *Signal Length*.
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp16s* *pDp, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer dot product method, return value is 16-bit signed short integer.

Parameters

- **pSrc1** – *Source Signal Pointer*.
- **pSrc2** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pDp** – Device memory pointer to the dot product result.
- **nScaleFactor** – *Integer Result Scaling*.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16s_Sfs_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16sc_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16sc_Sfs_Ctx.

Parameters

- **nLength** – *Signal Length*.
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16sc_Sfs_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp16sc* *pDp, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.

Parameters

- **pSrc1** – *Source Signal Pointer*.
- **pSrc2** – *Source Signal Pointer*.

- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16sc_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_32s_Sfs_Ctx**(const *Npp32s* *pSrc1, const *Npp32s* *pSrc2, size_t nLength, *Npp32s* *pDp, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit signed integer dot product method, return value is 32-bit signed integer.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_32sc_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_32sc_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_32sc_Sfs_Ctx**(const *Npp32sc* *pSrc1, const *Npp32sc* *pSrc2, size_t nLength, *Npp32sc* *pDp, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit signed integer complex dot product method, return value is 32-bit signed integer complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_32sc_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16s32s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp32s* *pDp, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer dot product method, return value is 32-bit signed integer.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.

- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16s32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16s16sc32sc_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16s16sc32sc_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16s16sc32sc_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp32sc* *pDp, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16s16sc32sc_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16s32s32s_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16s32s32s_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16s32s32s_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp32s* *pSrc2, size_t nLength, *Npp32s* *pDp, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer and 32-bit signed integer dot product method, return value is 32-bit signed integer.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **nScaleFactor** – *Integer Result Scaling*.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16s32s32s_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsDotProdGetBufferSize_16s16sc_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16s16sc_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16s16sc_Sfs_Ctx**(const *Npp16s* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp16sc* *pDp, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16s16sc_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsDotProdGetBufferSize_16sc32sc_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_16sc32sc_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_16sc32sc_Sfs_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp32sc* *pDp, int nScaleFactor, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_16sc32sc_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsDotProdGetBufferSize_32s32sc_Sfs_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsDotProd_32s32sc_Sfs_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsDotProd_32s32sc_Sfs_Ctx**(const *Npp32s* *pSrc1, const *Npp32sc* *pSrc2, size_t
 nLength, *Npp32sc* *pDp, int nScaleFactor, *Npp8u*
 *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit signed short integer and 32-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDp** – Device memory pointer to the dot product result.
- ▶ **nScaleFactor** – *Integer Result Scaling.*
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsDotProdGetBufferSize_32s32sc_Sfs_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

1.21.11. Signal Count In Range

1.21.11.1 Count In Range

Calculates the number of elements from specified range in the samples of a signal.

Functions

NppStatus **nppsCountInRangeGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsCountInRange_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsCountInRange_32s_Ctx**(const *Npp32s* *pSrc, size_t nLength, int *pCounts, *Npp32s* nLowerBound, *Npp32s* nUpperBound, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

Computes the number of elements whose values fall into the specified range on a 32-bit signed integer array.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pCounts** – Device memory pointer to the number of elements.
- ▶ **nLowerBound** – Lower bound of the specified range.
- ▶ **nUpperBound** – Upper bound of the specified range.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsCountInRangeGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.12. Signal Count Zero Crossings

1.21.12.1 Count Zero Crossings

Calculates the number of zero crossings in a signal.

Functions

NppStatus **nppsZeroCrossingGetBufferSize_16s32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsZeroCrossing_16s32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsZeroCrossing_16s32f_Ctx**(const *Npp16s* *pSrc, size_t nLength, *Npp32f* *pValZC, *NppsZCType* tZCType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer zero crossing method, return value is 32-bit floating point.

Parameters

- ▶ **pSrc** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pValZC** – Device memory pointer to the output result.
- ▶ **tZCType** – Type of the zero crossing measure: nppZCR, nppZCXor or nppZCC.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsZeroCrossingGetBufferSize_16s32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsZeroCrossingGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsZeroCrossing_32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsZeroCrossing_32f_Ctx**(const *Npp32f* *pSrc, size_t nLength, *Npp32f* *pValZC, *NppsZCType* tZCType, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit floating-point zero crossing method, return value is 32-bit floating point.

Parameters

- ▶ **pSrc** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pValZC** – Device memory pointer to the output result.
- ▶ **tZCType** – Type of the zero crossing measure: nppZCR, nppZCXor or nppZCC.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsZeroCrossingGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

1.21.13. Signal Maximum Error

1.21.13.1 MaximumError

Primitives for computing the maximum error between two signals. Given two signals *pSrc1* and *pSrc2* both with length *N*, the maximum error is defined as the largest absolute difference between the corresponding elements of two signals.

If the signal is in complex format, the absolute value of the complex number is used.

Functions

NppStatus **nppsMaximumError_8u_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

8-bit unsigned char maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_8u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_8s_Ctx**(const *Npp8s* *pSrc1, const *Npp8s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

8-bit signed char maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_8s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_16u_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit unsigned short integer maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_16u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_16s_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.

- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_16sc_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit unsigned short complex integer maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_16sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_32u_Ctx**(const *Npp32u* *pSrc1, const *Npp32u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit unsigned short integer maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_32u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_32s_Ctx**(const *Npp32s* *pSrc1, const *Npp32s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit signed short integer maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_32sc_Ctx**(const *Npp32sc* *pSrc1, const *Npp32sc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit unsigned short complex integer maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_32sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_64s_Ctx**(const *Npp64s* *pSrc1, const *Npp64s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit signed short integer maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.

- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_64s_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_64sc_Ctx**(const *Npp64sc* *pSrc1, const *Npp64sc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit unsigned short complex integer maximum method.

Parameters

- **pSrc1** – *Source Signal Pointer*.
- **pSrc2** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pDst** – Device memory pointer to the error result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_64sc_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit floating point maximum method.

Parameters

- **pSrc1** – *Source Signal Pointer*.
- **pSrc2** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pDst** – Device memory pointer to the error result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_32fc_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit floating point complex maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_32fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit floating point maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumError_64fc_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit floating point complex maximum method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumErrorGetBufferSize_64fc_Ctx* to determine the minimum number of bytes required.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumErrorGetBufferSize_8u_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_8u.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_8s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_8s.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_16u_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_16u.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_16s.

Parameters

- **nLength** – *Signal Length.*
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_16sc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_16sc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_32u_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_32u.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_32sc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_32sc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_64s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_64s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_64sc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_64sc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_32fc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_32fc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumErrorGetBufferSize_64fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumError_64fc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

1.21.14. Signal Average Error

1.21.14.1 AverageError

Primitives for computing the Average error between two signals. Given two signals *pSrc1* and *pSrc2* both with length *N*, the average error is defined as

$$AverageError = \frac{1}{N} \sum_{n=0}^{N-1} |pSrc1(n) - pSrc2(n)|$$

If the signal is in complex format, the absolute value of the complex number is used.

Functions

NppStatus **nppsAverageError_8u_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

8-bit unsigned char Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_8u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_8s_Ctx**(const *Npp8s* *pSrc1, const *Npp8s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

8-bit signed char Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_8s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_16u_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit unsigned short integer Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.

- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_16u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_16s_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_16sc_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit unsigned short complex integer Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_16sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_32u_Ctx**(const *Npp32u* *pSrc1, const *Npp32u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit unsigned short integer Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_32u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_32s_Ctx**(const *Npp32s* *pSrc1, const *Npp32s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit signed short integer Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_32sc_Ctx**(const *Npp32sc* *pSrc1, const *Npp32sc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit unsigned short complex integer Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.

- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_32sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_64s_Ctx**(const *Npp64s* *pSrc1, const *Npp64s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit signed short integer Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_64s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_64sc_Ctx**(const *Npp64sc* *pSrc1, const *Npp64sc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit unsigned short complex integer Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_64sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit floating point Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

```
NppStatus nppsAverageError_32fc_Ctx(const Npp32fc *pSrc1, const Npp32fc *pSrc2, size_t  
nLength, Npp64f *pDst, Npp8u *pDeviceBuffer,  
NppStreamContext nppStreamCtx)
```

32-bit floating point complex Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_32fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

```
NppStatus nppsAverageError_64f_Ctx(const Npp64f *pSrc1, const Npp64f *pSrc2, size_t  
nLength, Npp64f *pDst, Npp8u *pDeviceBuffer,  
NppStreamContext nppStreamCtx)
```

64-bit floating point Average method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.

- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageError_64fc_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit floating point complex Average method.

Parameters

- **pSrc1** – *Source Signal Pointer*.
- **pSrc2** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pDst** – Device memory pointer to the error result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageErrorGetBufferSize_64fc_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageErrorGetBufferSize_8u_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_8u.

Parameters

- **nLength** – *Signal Length*.
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_8s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_8s.

Parameters

- **nLength** – *Signal Length*.
- **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_16u_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_16u.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_16s_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_16s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_16sc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_16sc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_32u_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_32u.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_32sc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_32sc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_64s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_64s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_64sc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_64sc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_32fc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_32fc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageErrorGetBufferSize_64fc_Ctx**(size_t nLength, size_t *hpBufferSize,
NppStreamContext nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageError_64fc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

1.21.15. Signal Maximum Relative Error

1.21.15.1 MaximumRelativeError

Primitives for computing the MaximumRelative error between two signals. Given two signals $pSrc1$ and $pSrc2$ both with length N , the maximum relative error is defined as

$$MaximumRelativeError = \max \frac{|pSrc1(n) - pSrc2(n)|}{\max(|pSrc1(n)|, |pSrc2(n)|)}$$

If the signal is in complex format, the absolute value of the complex number is used.

Functions

NppStatus **nppsMaximumRelativeError_8u_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

8-bit unsigned char MaximumRelative method.

Parameters

- **pSrc1** – *Source Signal Pointer*.
- **pSrc2** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pDst** – Device memory pointer to the error result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumRelativeErrorGetBufferSize_8u_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumRelativeError_8s_Ctx**(const *Npp8s* *pSrc1, const *Npp8s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

8-bit signed char MaximumRelative method.

Parameters

- **pSrc1** – *Source Signal Pointer*.
- **pSrc2** – *Source Signal Pointer*.
- **nLength** – *Signal Length*.
- **pDst** – Device memory pointer to the error result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumRelativeErrorGetBufferSize_8s_Ctx* to determine the minimum number of bytes required.

- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumRelativeError_16u_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit unsigned short integer MaximumRelative method.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer.*
- **nLength** – *Signal Length.*
- **pDst** – Device memory pointer to the error result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumRelativeErrorGetBufferSize_16u_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumRelativeError_16s_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer MaximumRelative method.

Parameters

- **pSrc1** – *Source Signal Pointer.*
- **pSrc2** – *Source Signal Pointer.*
- **nLength** – *Signal Length.*
- **pDst** – Device memory pointer to the error result.
- **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumRelativeErrorGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumRelativeError_16sc_Ctx**(const *Npp16sc* *pSrc1, const *Npp16sc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit unsigned short complex integer MaximumRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMaximumRelativeErrorGetBufferSize_16sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumRelativeError_32u_Ctx**(const *Npp32u* *pSrc1, const *Npp32u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit unsigned short integer MaximumRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMaximumRelativeErrorGetBufferSize_32u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumRelativeError_32s_Ctx**(const *Npp32s* *pSrc1, const *Npp32s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit signed short integer MaximumRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.

- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumRelativeErrorGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

```
NppStatus nppsMaximumRelativeError_32sc_Ctx(const Npp32sc *pSrc1, const Npp32sc *pSrc2,  
                                             size_t nLength, Npp64f *pDst, Npp8u  
                                             *pDeviceBuffer, NppStreamContext  
                                             nppStreamCtx)
```

32-bit unsigned short complex integer MaximumRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumRelativeErrorGetBufferSize_32sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

```
NppStatus nppsMaximumRelativeError_64s_Ctx(const Npp64s *pSrc1, const Npp64s *pSrc2,  
                                             size_t nLength, Npp64f *pDst, Npp8u  
                                             *pDeviceBuffer, NppStreamContext  
                                             nppStreamCtx)
```

64-bit signed short integer MaximumRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumRelativeErrorGetBufferSize_64s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

```
NppStatus nppsMaximumRelativeError_64sc_Ctx(const Npp64sc *pSrc1, const Npp64sc *pSrc2,
                                             size_t nLength, Npp64f *pDst, Npp8u
                                             *pDeviceBuffer, NppStreamContext
                                             nppStreamCtx)
```

64-bit unsigned short complex integer MaximumRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMaximumRelativeErrorGetBufferSize_64sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

```
NppStatus nppsMaximumRelativeError_32f_Ctx(const Npp32f *pSrc1, const Npp32f *pSrc2,
                                             size_t nLength, Npp64f *pDst, Npp8u
                                             *pDeviceBuffer, NppStreamContext
                                             nppStreamCtx)
```

32-bit floating point MaximumRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsMaximumRelativeErrorGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

```
NppStatus nppsMaximumRelativeError_32fc_Ctx(const Npp32fc *pSrc1, const Npp32fc *pSrc2,
                                              size_t nLength, Npp64f *pDst, Npp8u
                                              *pDeviceBuffer, NppStreamContext
                                              nppStreamCtx)
```

32-bit floating point complex MaximumRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*

- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumRelativeErrorGetBufferSize_32fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumRelativeError_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit floating point MaximumRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumRelativeErrorGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsMaximumRelativeError_64fc_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit floating point complex MaximumRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsMaximumRelativeErrorGetBufferSize_64fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsMaximumRelativeErrorGetBufferSize_8u_Ctx**(size_t nLength, size_t
 *hpBufferSize,
 NppStreamContext
 nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_8u.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_8s_Ctx**(size_t nLength, size_t
 *hpBufferSize,
 NppStreamContext
 nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_8s.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_16u_Ctx**(size_t nLength, size_t
 *hpBufferSize,
 NppStreamContext
 nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_16u.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_16s_Ctx**(size_t nLength, size_t
 *hpBufferSize,
 NppStreamContext
 nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_16s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_16sc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_16sc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_32u_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_32u.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_32s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_32sc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_32sc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_64s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_64s.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_64sc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_64sc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_32f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_32fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_32fc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_64f.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsMaximumRelativeErrorGetBufferSize_64fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsMaximumRelativeError_64fc.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

1.21.16. Signal Average Relative Error

1.21.16.1 AverageRelativeError

Primitives for computing the AverageRelative error between two signals. Given two signals *pSrc1* and *pSrc2* both with length *N*, the average relative error is defined as

$$\text{AverageRelativeError} = \frac{1}{N} \sum_{n=0}^{N-1} \frac{|pSrc1(n) - pSrc2(n)|}{\max(|pSrc1(n)|, |pSrc2(n)|)}$$

If the signal is in complex format, the absolute value of the complex number is used.

Functions

NppStatus **nppsAverageRelativeError_8u_Ctx**(const *Npp8u* *pSrc1, const *Npp8u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

8-bit unsigned char AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_8u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageRelativeError_8s_Ctx**(const *Npp8s* *pSrc1, const *Npp8s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

8-bit signed char AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.

- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_8s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageRelativeError_16u_Ctx**(const *Npp16u* *pSrc1, const *Npp16u* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit unsigned short integer AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_16u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageRelativeError_16s_Ctx**(const *Npp16s* *pSrc1, const *Npp16s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

16-bit signed short integer AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_16s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

```
NppStatus nppsAverageRelativeError_16sc_Ctx(const Npp16sc *pSrc1, const Npp16sc *pSrc2,
                                             size_t nLength, Npp64f *pDst, Npp8u
                                             *pDeviceBuffer, NppStreamContext
                                             nppStreamCtx)
```

16-bit unsigned short complex integer AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsAverageRelativeErrorGetBufferSize_16sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

```
NppStatus nppsAverageRelativeError_32u_Ctx(const Npp32u *pSrc1, const Npp32u *pSrc2,
                                             size_t nLength, Npp64f *pDst, Npp8u
                                             *pDeviceBuffer, NppStreamContext
                                             nppStreamCtx)
```

32-bit unsigned short integer AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer.* Use *nppsAverageRelativeErrorGetBufferSize_32u_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

```
NppStatus nppsAverageRelativeError_32s_Ctx(const Npp32s *pSrc1, const Npp32s *pSrc2,
                                             size_t nLength, Npp64f *pDst, Npp8u
                                             *pDeviceBuffer, NppStreamContext
                                             nppStreamCtx)
```

32-bit signed short integer AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*

- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_32s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageRelativeError_32sc_Ctx**(const *Npp32sc* *pSrc1, const *Npp32sc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit unsigned short complex integer AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_32sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageRelativeError_64s_Ctx**(const *Npp64s* *pSrc1, const *Npp64s* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit signed short integer AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer*.
- ▶ **pSrc2** – *Source Signal Pointer*.
- ▶ **nLength** – *Signal Length*.
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_64s_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageRelativeError_64sc_Ctx**(const *Npp64sc* *pSrc1, const *Npp64sc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit unsigned short complex integer AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_64sc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageRelativeError_32f_Ctx**(const *Npp32f* *pSrc1, const *Npp32f* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit floating point AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_32f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageRelativeError_32fc_Ctx**(const *Npp32fc* *pSrc1, const *Npp32fc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

32-bit floating point complex AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_32fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageRelativeError_64f_Ctx**(const *Npp64f* *pSrc1, const *Npp64f* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit floating point AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_64f_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

Signal Data Related Error Codes, Length Related Error Codes.

NppStatus **nppsAverageRelativeError_64fc_Ctx**(const *Npp64fc* *pSrc1, const *Npp64fc* *pSrc2, size_t nLength, *Npp64f* *pDst, *Npp8u* *pDeviceBuffer, *NppStreamContext* nppStreamCtx)

64-bit floating point complex AverageRelative method.

Parameters

- ▶ **pSrc1** – *Source Signal Pointer.*
- ▶ **pSrc2** – *Source Signal Pointer.*
- ▶ **nLength** – *Signal Length.*
- ▶ **pDst** – Device memory pointer to the error result.
- ▶ **pDeviceBuffer** – Pointer to the required device memory allocation, *Scratch Buffer and Host Pointer*. Use *nppsAverageRelativeErrorGetBufferSize_64fc_Ctx* to determine the minimum number of bytes required.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns*Signal Data Related Error Codes, Length Related Error Codes.*

NppStatus **nppsAverageRelativeErrorGetBufferSize_8u_Ctx**(size_t nLength, size_t
 *hpBufferSize,
 NppStreamContext
 nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_8u_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_8s_Ctx**(size_t nLength, size_t
 *hpBufferSize,
 NppStreamContext
 nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_8s_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_16u_Ctx**(size_t nLength, size_t
 *hpBufferSize,
 NppStreamContext
 nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_16u_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_16s_Ctx**(size_t nLength, size_t
 *hpBufferSize,
 NppStreamContext
 nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_16s_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_16sc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_16sc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_32u_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_32u_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_32s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_32s_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_32sc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_32sc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_64s_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_64s_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_64sc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_64sc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length.*
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context.*

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_32f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_32f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_32fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_32fc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_64f_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_64f_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns

NPP_SUCCESS

NppStatus **nppsAverageRelativeErrorGetBufferSize_64fc_Ctx**(size_t nLength, size_t *hpBufferSize, *NppStreamContext* nppStreamCtx)

Device-buffer size (in bytes) for nppsAverageRelativeError_64fc_Ctx.

Parameters

- ▶ **nLength** – *Signal Length*.
- ▶ **hpBufferSize** – Required buffer size. Important: hpBufferSize is a *host pointer*.
- ▶ **nppStreamCtx** – *Application Managed Stream Context*.

Returns
NPP_SUCCESS

1.22. Signal Memory Management Functions

Functions that provide memory management functionality like malloc and free.

1.22.1. Malloc

Signal-allocator methods for allocating 1D arrays of data in device memory. All allocators have size parameters to specify the size of the signal (1D array) being allocated.

The allocator methods return a pointer to the newly allocated memory of appropriate type. If device-memory allocation is not possible due to resource constraints the allocators return 0 (i.e. NULL pointer).

All signal allocators allocate memory aligned such that it is beneficial to the performance of the majority of the signal-processing primitives. It is no mandatory however to use these allocators. Any valid CUDA device-memory pointers can be passed to NPP primitives.

Functions

Npp8u ***nppsMalloc_8u**(size_t nSize)
8-bit unsigned signal allocator.

Parameters

- **nSize** – Number of unsigned chars in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp8s ***nppsMalloc_8s**(size_t nSize)
8-bit signed signal allocator.

Parameters

- **nSize** – Number of (signed) chars in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp16u ***nppsMalloc_16u**(size_t nSize)
16-bit unsigned signal allocator.

Parameters

- **nSize** – Number of unsigned shorts in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp16s *nppsMalloc_16s(size_t nSize)

16-bit signal allocator.

Parameters

- **nSize** – Number of shorts in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp16sc *nppsMalloc_16sc(size_t nSize)

16-bit complex-value signal allocator.

Parameters

- **nSize** – Number of 16-bit complex numbers in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp32u *nppsMalloc_32u(size_t nSize)

32-bit unsigned signal allocator.

Parameters

- **nSize** – Number of unsigned ints in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp32s *nppsMalloc_32s(size_t nSize)

32-bit integer signal allocator.

Parameters

- **nSize** – Number of ints in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp32sc *nppsMalloc_32sc(size_t nSize)

32-bit complex integer signal allocator.

Parameters

- **nSize** – Number of complex integer values in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp32f *nppsMalloc_32f(size_t nSize)

32-bit float signal allocator.

Parameters

- **nSize** – Number of floats in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp32fc *nppsMalloc_32fc(size_t nSize)

32-bit complex float signal allocator.

Parameters

- **nSize** – Number of complex float values in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp64s *nppsMalloc_64s(size_t nSize)

64-bit long integer signal allocator.

Parameters

- **nSize** – Number of long ints in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp64sc *nppsMalloc_64sc(size_t nSize)

64-bit complex long integer signal allocator.

Parameters

- **nSize** – Number of complex long int values in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp64f *nppsMalloc_64f(size_t nSize)

64-bit float (double) signal allocator.

Parameters

- **nSize** – Number of doubles in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

Npp64fc *nppsMalloc_64fc(size_t nSize)

64-bit complex complex signal allocator.

Parameters

- **nSize** – Number of complex double values in the new signal.

Returns

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

1.22.2. Free

Free signal memory.

Functions

void **nppsFree**(void *pValues)

Free method for any signal memory.

Parameters

- **pValues** – A pointer to memory allocated using `nppiMalloc_<modifier>`.

Copyright

©2009-2026, NVIDIA Corporation & affiliates. All rights reserved

Index

N

Npp8s (C type), 28
Npp8u (C type), 28
Npp8uc (C struct), 28
Npp8uc.im (C var), 29
Npp8uc.re (C var), 29
Npp16f (C struct), 20
Npp16f_2 (C struct), 20
Npp16f_2.fp16_0 (C var), 21
Npp16f_2.fp16_1 (C var), 21
Npp16f.fp16 (C var), 20
Npp16s (C type), 28
Npp16sc (C struct), 29
Npp16sc.im (C var), 29
Npp16sc.re (C var), 29
Npp16u (C type), 28
Npp16uc (C struct), 29
Npp16uc.im (C var), 29
Npp16uc.re (C var), 29
Npp32f (C type), 28
Npp32fc (C struct), 30
Npp32fc.im (C var), 30
Npp32fc.re (C var), 30
Npp32s (C type), 28
Npp32sc (C struct), 29
Npp32sc.im (C var), 30
Npp32sc.re (C var), 30
Npp32u (C type), 28
Npp32uc (C struct), 29
Npp32uc.im (C var), 29
Npp32uc.re (C var), 29
Npp64f (C type), 28
Npp64fc (C struct), 30
Npp64fc.im (C var), 30
Npp64fc.re (C var), 30
Npp64s (C type), 28
Npp64sc (C struct), 30
Npp64sc.im (C var), 30
Npp64sc.re (C var), 30
Npp64u (C type), 28
NPP_CONTOUR_DIRECTION_ANY_EAST (C macro), 45
NPP_CONTOUR_DIRECTION_ANY_NORTH (C macro), 45
NPP_CONTOUR_DIRECTION_ANY_SOUTH (C macro), 45
NPP_CONTOUR_DIRECTION_ANY_WEST (C macro), 45
NPP_CONTOUR_DIRECTION_EAST (C macro), 44
NPP_CONTOUR_DIRECTION_NORTH (C macro), 44
NPP_CONTOUR_DIRECTION_NORTH_EAST (C macro), 44
NPP_CONTOUR_DIRECTION_NORTH_WEST (C macro), 44
NPP_CONTOUR_DIRECTION_SOUTH (C macro), 44
NPP_CONTOUR_DIRECTION_SOUTH_EAST (C macro), 44
NPP_CONTOUR_DIRECTION_SOUTH_WEST (C macro), 44
NPP_CONTOUR_DIRECTION_WEST (C macro), 44
NPP_HOG_MAX_BINS_PER_CELL (C macro), 39
NPP_HOG_MAX_BLOCK_SIZE (C macro), 39
NPP_HOG_MAX_CELL_SIZE (C macro), 39
NPP_HOG_MAX_CELLS_PER_DESCRIPTOR (C macro), 39
NPP_HOG_MAX_DESCRIPTOR_LOCATIONS_PER_CALL (C macro), 39
NPP_HOG_MAX_OVERLAPPING_BLOCKS_PER_DESCRIPTOR (C macro), 39
NPP_MAX_8S (C macro), 33
NPP_MAX_8U (C macro), 32
NPP_MAX_16S (C macro), 33
NPP_MAX_16U (C macro), 32
NPP_MAX_32S (C macro), 33
NPP_MAX_32U (C macro), 32
NPP_MAX_64S (C macro), 33
NPP_MAX_64U (C macro), 32
NPP_MAXABS_32F (C macro), 33
NPP_MAXABS_64F (C macro), 33
NPP_MIN_8S (C macro), 32
NPP_MIN_8U (C macro), 32
NPP_MIN_16S (C macro), 33
NPP_MIN_16U (C macro), 32
NPP_MIN_32S (C macro), 33
NPP_MIN_32U (C macro), 32

NPP_MIN_64S (C macro), 33
 NPP_MIN_64U (C macro), 32
 NPP_MINABS_32F (C macro), 33
 NPP_MINABS_64F (C macro), 33
 NppCmpOp (C enum), 35
 NppCmpOp.NPP_CMP_EQ (C enumerator), 36
 NppCmpOp.NPP_CMP_GREATER (C enumerator), 36
 NppCmpOp.NPP_CMP_GREATER_EQ (C enumerator), 36
 NppCmpOp.NPP_CMP_LESS (C enumerator), 35
 NppCmpOp.NPP_CMP_LESS_EQ (C enumerator), 36
 NppDataType (C enum), 30
 NppDataType.NPP_8S (C enumerator), 31
 NppDataType.NPP_8U (C enumerator), 31
 NppDataType.NPP_16F (C enumerator), 31
 NppDataType.NPP_16S (C enumerator), 31
 NppDataType.NPP_16U (C enumerator), 31
 NppDataType.NPP_32F (C enumerator), 31
 NppDataType.NPP_32S (C enumerator), 31
 NppDataType.NPP_32U (C enumerator), 31
 NppDataType.NPP_64F (C enumerator), 31
 NppDataType.NPP_64S (C enumerator), 31
 NppDataType.NPP_64U (C enumerator), 31
 nppGetLibVersion (C function), 52
 NppHintAlgorithm (C enum), 37
 NppHintAlgorithm.NPP_ALG_HINT_ACCURATE (C enumerator), 38
 NppHintAlgorithm.NPP_ALG_HINT_FAST (C enumerator), 38
 NppHintAlgorithm.NPP_ALG_HINT_NONE (C enumerator), 38
 nppiAbs_16f_C1IR_Ctx (C function), 369
 nppiAbs_16f_C1R_Ctx (C function), 369
 nppiAbs_16f_C3IR_Ctx (C function), 370
 nppiAbs_16f_C3R_Ctx (C function), 369
 nppiAbs_16f_C4IR_Ctx (C function), 370
 nppiAbs_16f_C4R_Ctx (C function), 370
 nppiAbs_16s_AC4IR_Ctx (C function), 368
 nppiAbs_16s_AC4R_Ctx (C function), 367
 nppiAbs_16s_C1IR_Ctx (C function), 367
 nppiAbs_16s_C1R_Ctx (C function), 366
 nppiAbs_16s_C3IR_Ctx (C function), 367
 nppiAbs_16s_C3R_Ctx (C function), 367
 nppiAbs_16s_C4IR_Ctx (C function), 368
 nppiAbs_16s_C4R_Ctx (C function), 368
 nppiAbs_32f_AC4IR_Ctx (C function), 372
 nppiAbs_32f_AC4R_Ctx (C function), 372
 nppiAbs_32f_C1IR_Ctx (C function), 371
 nppiAbs_32f_C1R_Ctx (C function), 370
 nppiAbs_32f_C3IR_Ctx (C function), 371
 nppiAbs_32f_C3R_Ctx (C function), 371
 nppiAbs_32f_C4IR_Ctx (C function), 373
 nppiAbs_32f_C4R_Ctx (C function), 372
 nppiAbsDiff_8u_C1R_Ctx (C function), 373
 nppiAbsDiff_8u_C4R_Ctx (C function), 374
 nppiAbsDiff_16f_C1R_Ctx (C function), 375
 nppiAbsDiff_16u_C1R_Ctx (C function), 374
 nppiAbsDiff_32f_C1R_Ctx (C function), 375
 nppiAbsDiffC_8u_C1R_Ctx (C function), 229
 nppiAbsDiffC_16u_C1R_Ctx (C function), 229
 nppiAbsDiffC_32f_C1R_Ctx (C function), 230
 nppiAbsDiffDeviceC_8u_C1R_Ctx (C function), 229
 nppiAbsDiffDeviceC_16u_C1R_Ctx (C function), 230
 nppiAbsDiffDeviceC_32f_C1R_Ctx (C function), 231
 nppiAdd_8u_AC4IRSfs_Ctx (C function), 233
 nppiAdd_8u_AC4RSfs_Ctx (C function), 233
 nppiAdd_8u_C1IRSfs_Ctx (C function), 232
 nppiAdd_8u_C1RSfs_Ctx (C function), 231
 nppiAdd_8u_C3IRSfs_Ctx (C function), 232
 nppiAdd_8u_C3RSfs_Ctx (C function), 232
 nppiAdd_8u_C4IRSfs_Ctx (C function), 234
 nppiAdd_8u_C4RSfs_Ctx (C function), 234
 nppiAdd_16f_C1IR_Ctx (C function), 250
 nppiAdd_16f_C1R_Ctx (C function), 250
 nppiAdd_16f_C3IR_Ctx (C function), 251
 nppiAdd_16f_C3R_Ctx (C function), 250
 nppiAdd_16f_C4IR_Ctx (C function), 251
 nppiAdd_16f_C4R_Ctx (C function), 251
 nppiAdd_16s_AC4IRSfs_Ctx (C function), 240
 nppiAdd_16s_AC4RSfs_Ctx (C function), 240
 nppiAdd_16s_C1IRSfs_Ctx (C function), 239
 nppiAdd_16s_C1RSfs_Ctx (C function), 238
 nppiAdd_16s_C3IRSfs_Ctx (C function), 239
 nppiAdd_16s_C3RSfs_Ctx (C function), 239
 nppiAdd_16s_C4IRSfs_Ctx (C function), 241
 nppiAdd_16s_C4RSfs_Ctx (C function), 241
 nppiAdd_16sc_AC4IRSfs_Ctx (C function), 244
 nppiAdd_16sc_AC4RSfs (C function), 244
 nppiAdd_16sc_AC4RSfs_Ctx (C function), 243
 nppiAdd_16sc_C1IRSfs_Ctx (C function), 242
 nppiAdd_16sc_C1RSfs_Ctx (C function), 242
 nppiAdd_16sc_C3IRSfs_Ctx (C function), 243
 nppiAdd_16sc_C3RSfs_Ctx (C function), 243
 nppiAdd_16u_AC4IRSfs_Ctx (C function), 237
 nppiAdd_16u_AC4RSfs_Ctx (C function), 236
 nppiAdd_16u_C1IRSfs_Ctx (C function), 235
 nppiAdd_16u_C1RSfs_Ctx (C function), 235
 nppiAdd_16u_C3IRSfs_Ctx (C function), 236
 nppiAdd_16u_C3RSfs_Ctx (C function), 235
 nppiAdd_16u_C4IRSfs_Ctx (C function), 238
 nppiAdd_16u_C4RSfs_Ctx (C function), 237
 nppiAdd_32f_AC4IR_Ctx (C function), 254
 nppiAdd_32f_AC4R_Ctx (C function), 253
 nppiAdd_32f_C1IR_Ctx (C function), 252

nppiAdd_32f_C1R_Ctx (C function), 252
 nppiAdd_32f_C3IR_Ctx (C function), 253
 nppiAdd_32f_C3R_Ctx (C function), 253
 nppiAdd_32f_C4IR_Ctx (C function), 255
 nppiAdd_32f_C4R_Ctx (C function), 254
 nppiAdd_32fc_AC4IR_Ctx (C function), 257
 nppiAdd_32fc_AC4R_Ctx (C function), 256
 nppiAdd_32fc_C1IR_Ctx (C function), 255
 nppiAdd_32fc_C1R_Ctx (C function), 255
 nppiAdd_32fc_C3IR_Ctx (C function), 256
 nppiAdd_32fc_C3R_Ctx (C function), 256
 nppiAdd_32fc_C4IR_Ctx (C function), 258
 nppiAdd_32fc_C4R_Ctx (C function), 257
 nppiAdd_32s_C1IRSfs_Ctx (C function), 246
 nppiAdd_32s_C1R_Ctx (C function), 245
 nppiAdd_32s_C1RSfs_Ctx (C function), 245
 nppiAdd_32s_C3IRSfs_Ctx (C function), 247
 nppiAdd_32s_C3RSfs_Ctx (C function), 246
 nppiAdd_32sc_AC4IRSfs_Ctx (C function), 249
 nppiAdd_32sc_AC4RSfs_Ctx (C function), 249
 nppiAdd_32sc_C1IRSfs_Ctx (C function), 247
 nppiAdd_32sc_C1RSfs_Ctx (C function), 247
 nppiAdd_32sc_C3IRSfs_Ctx (C function), 248
 nppiAdd_32sc_C3RSfs_Ctx (C function), 248
 nppiAddC_8u_AC4IRSfs_Ctx (C function), 56
 nppiAddC_8u_AC4RSfs_Ctx (C function), 55
 nppiAddC_8u_C1IRSfs_Ctx (C function), 53
 nppiAddC_8u_C1RSfs_Ctx (C function), 52
 nppiAddC_8u_C3IRSfs_Ctx (C function), 55
 nppiAddC_8u_C3RSfs_Ctx (C function), 54
 nppiAddC_8u_C4IRSfs_Ctx (C function), 58
 nppiAddC_8u_C4RSfs_Ctx (C function), 57
 nppiAddC_16f_C1IR_Ctx (C function), 80
 nppiAddC_16f_C1R_Ctx (C function), 80
 nppiAddC_16f_C3IR_Ctx (C function), 82
 nppiAddC_16f_C3R_Ctx (C function), 81
 nppiAddC_16f_C4IR_Ctx (C function), 83
 nppiAddC_16f_C4R_Ctx (C function), 83
 nppiAddC_16s_AC4IRSfs_Ctx (C function), 69
 nppiAddC_16s_AC4RSfs_Ctx (C function), 68
 nppiAddC_16s_C1IRSfs_Ctx (C function), 66
 nppiAddC_16s_C1RSfs_Ctx (C function), 65
 nppiAddC_16s_C3IRSfs_Ctx (C function), 68
 nppiAddC_16s_C3RSfs_Ctx (C function), 67
 nppiAddC_16s_C4IRSfs_Ctx (C function), 71
 nppiAddC_16s_C4RSfs_Ctx (C function), 70
 nppiAddC_16sc_AC4IRSfs_Ctx (C function), 74
 nppiAddC_16sc_AC4RSfs_Ctx (C function), 73
 nppiAddC_16sc_C1IRSfs_Ctx (C function), 72
 nppiAddC_16sc_C1RSfs_Ctx (C function), 72
 nppiAddC_16sc_C3IRSfs_Ctx (C function), 73
 nppiAddC_16sc_C3RSfs_Ctx (C function), 73
 nppiAddC_16u_AC4IRSfs_Ctx (C function), 63
 nppiAddC_16u_AC4RSfs_Ctx (C function), 62
 nppiAddC_16u_C1IRSfs_Ctx (C function), 60
 nppiAddC_16u_C1RSfs_Ctx (C function), 59
 nppiAddC_16u_C3IRSfs_Ctx (C function), 61
 nppiAddC_16u_C3RSfs_Ctx (C function), 60
 nppiAddC_16u_C4IRSfs_Ctx (C function), 65
 nppiAddC_16u_C4RSfs_Ctx (C function), 64
 nppiAddC_32f_AC4IR_Ctx (C function), 88
 nppiAddC_32f_AC4R_Ctx (C function), 87
 nppiAddC_32f_C1IR_Ctx (C function), 85
 nppiAddC_32f_C1R_Ctx (C function), 84
 nppiAddC_32f_C3IR_Ctx (C function), 86
 nppiAddC_32f_C3R_Ctx (C function), 85
 nppiAddC_32f_C4IR_Ctx (C function), 89
 nppiAddC_32f_C4R_Ctx (C function), 88
 nppiAddC_32fc_AC4IR_Ctx (C function), 92
 nppiAddC_32fc_AC4R_Ctx (C function), 91
 nppiAddC_32fc_C1IR_Ctx (C function), 90
 nppiAddC_32fc_C1R_Ctx (C function), 90
 nppiAddC_32fc_C3IR_Ctx (C function), 91
 nppiAddC_32fc_C3R_Ctx (C function), 90
 nppiAddC_32fc_C4IR_Ctx (C function), 92
 nppiAddC_32fc_C4R_Ctx (C function), 92
 nppiAddC_32s_C1IRSfs_Ctx (C function), 75
 nppiAddC_32s_C1RSfs_Ctx (C function), 74
 nppiAddC_32s_C3IRSfs_Ctx (C function), 77
 nppiAddC_32s_C3RSfs_Ctx (C function), 76
 nppiAddC_32sc_AC4IRSfs_Ctx (C function), 79
 nppiAddC_32sc_AC4RSfs_Ctx (C function), 79
 nppiAddC_32sc_C1IRSfs_Ctx (C function), 78
 nppiAddC_32sc_C1RSfs_Ctx (C function), 77
 nppiAddC_32sc_C3IRSfs_Ctx (C function), 79
 nppiAddC_32sc_C3RSfs_Ctx (C function), 78
 nppiAddDeviceC_8u_AC4IRSfs_Ctx (C function), 57
 nppiAddDeviceC_8u_AC4RSfs_Ctx (C function), 56
 nppiAddDeviceC_8u_C1IRSfs_Ctx (C function), 53
 nppiAddDeviceC_8u_C1RSfs_Ctx (C function), 53
 nppiAddDeviceC_8u_C3IRSfs_Ctx (C function), 55
 nppiAddDeviceC_8u_C3RSfs_Ctx (C function), 54
 nppiAddDeviceC_8u_C4IRSfs_Ctx (C function), 58
 nppiAddDeviceC_8u_C4RSfs_Ctx (C function), 58
 nppiAddDeviceC_16f_C1IR_Ctx (C function), 81
 nppiAddDeviceC_16f_C1R_Ctx (C function), 80
 nppiAddDeviceC_16f_C3IR_Ctx (C function), 82
 nppiAddDeviceC_16f_C3R_Ctx (C function), 82

nppiAddDeviceC_16f_C4IR_Ctx (C function), 84
 nppiAddDeviceC_16f_C4R_Ctx (C function), 83
 nppiAddDeviceC_16s_AC4IRSfs_Ctx (C function), 70
 nppiAddDeviceC_16s_AC4RSfs_Ctx (C function), 69
 nppiAddDeviceC_16s_C1IRSfs_Ctx (C function), 66
 nppiAddDeviceC_16s_C1RSfs_Ctx (C function), 66
 nppiAddDeviceC_16s_C3IRSfs_Ctx (C function), 68
 nppiAddDeviceC_16s_C3RSfs_Ctx (C function), 67
 nppiAddDeviceC_16s_C4IRSfs_Ctx (C function), 71
 nppiAddDeviceC_16s_C4RSfs_Ctx (C function), 71
 nppiAddDeviceC_16u_AC4IRSfs_Ctx (C function), 63
 nppiAddDeviceC_16u_AC4RSfs_Ctx (C function), 62
 nppiAddDeviceC_16u_C1IRSfs_Ctx (C function), 60
 nppiAddDeviceC_16u_C1RSfs_Ctx (C function), 59
 nppiAddDeviceC_16u_C3IRSfs_Ctx (C function), 62
 nppiAddDeviceC_16u_C3RSfs_Ctx (C function), 61
 nppiAddDeviceC_16u_C4IRSfs_Ctx (C function), 65
 nppiAddDeviceC_16u_C4RSfs_Ctx (C function), 64
 nppiAddDeviceC_32f_AC4IR_Ctx (C function), 88
 nppiAddDeviceC_32f_AC4R_Ctx (C function), 87
 nppiAddDeviceC_32f_C1IR_Ctx (C function), 85
 nppiAddDeviceC_32f_C1R_Ctx (C function), 84
 nppiAddDeviceC_32f_C3IR_Ctx (C function), 86
 nppiAddDeviceC_32f_C3R_Ctx (C function), 86
 nppiAddDeviceC_32f_C4IR_Ctx (C function), 89
 nppiAddDeviceC_32f_C4R_Ctx (C function), 89
 nppiAddDeviceC_32s_C1IRSfs_Ctx (C function), 75
 nppiAddDeviceC_32s_C1RSfs_Ctx (C function), 75
 nppiAddDeviceC_32s_C3IRSfs_Ctx (C function), 77
 nppiAddDeviceC_32s_C3RSfs_Ctx (C function), 76
 nppiAddProduct_8u32f_C1IMR_Ctx (C function), 261
 nppiAddProduct_8u32f_C1IR_Ctx (C function), 261
 nppiAddProduct_16f_C1IR_Ctx (C function), 264
 nppiAddProduct_16u32f_C1IMR_Ctx (C function), 262
 nppiAddProduct_16u32f_C1IR_Ctx (C function), 262
 nppiAddProduct_32f_C1IMR_Ctx (C function), 263
 nppiAddProduct_32f_C1IR_Ctx (C function), 263
 nppiAddSquare_8u32f_C1IMR_Ctx (C function), 258
 nppiAddSquare_8u32f_C1IR_Ctx (C function), 259
 nppiAddSquare_16u32f_C1IMR_Ctx (C function), 259
 nppiAddSquare_16u32f_C1IR_Ctx (C function), 259
 nppiAddSquare_32f_C1IMR_Ctx (C function), 260
 nppiAddSquare_32f_C1IR_Ctx (C function), 260
 nppiAddWeighted_8u32f_C1IMR_Ctx (C function), 264
 nppiAddWeighted_8u32f_C1IR_Ctx (C function), 265
 nppiAddWeighted_16u32f_C1IMR_Ctx (C function), 265
 nppiAddWeighted_16u32f_C1IR_Ctx (C function), 266
 nppiAddWeighted_32f_C1IMR_Ctx (C function), 266
 nppiAddWeighted_32f_C1IR_Ctx (C function), 266
 nppiAlphaComp_8s_AC1R_Ctx (C function), 506
 nppiAlphaComp_8u_AC1R_Ctx (C function), 505
 nppiAlphaComp_8u_AC4R_Ctx (C function), 505
 nppiAlphaComp_16s_AC1R_Ctx (C function), 507
 nppiAlphaComp_16u_AC1R_Ctx (C function), 506
 nppiAlphaComp_16u_AC4R_Ctx (C function), 506
 nppiAlphaComp_32f_AC1R_Ctx (C function), 509
 nppiAlphaComp_32f_AC4R_Ctx (C function), 510
 nppiAlphaComp_32s_AC1R_Ctx (C function),

508
 nppiAlphaComp_32s_AC4R_Ctx (C function),
 509
 nppiAlphaComp_32u_AC1R_Ctx (C function),
 507
 nppiAlphaComp_32u_AC4R_Ctx (C function),
 508
 nppiAlphaCompC_8s_C1R_Ctx (C function), 494
 nppiAlphaCompC_8u_AC4R_Ctx (C function),
 493
 nppiAlphaCompC_8u_C1R_Ctx (C function), 492
 nppiAlphaCompC_8u_C3R_Ctx (C function), 492
 nppiAlphaCompC_8u_C4R_Ctx (C function), 493
 nppiAlphaCompC_16s_C1R_Ctx (C function),
 496
 nppiAlphaCompC_16u_AC4R_Ctx (C function),
 496
 nppiAlphaCompC_16u_C1R_Ctx (C function),
 494
 nppiAlphaCompC_16u_C3R_Ctx (C function),
 495
 nppiAlphaCompC_16u_C4R_Ctx (C function),
 495
 nppiAlphaCompC_32f_C1R_Ctx (C function),
 498
 nppiAlphaCompC_32s_C1R_Ctx (C function),
 497
 nppiAlphaCompC_32u_C1R_Ctx (C function),
 497
 nppiAlphaCompColorKey_8u_AC4R_Ctx (C
 function), 555
 NppiAlphaOp (C enum), 38
 NppiAlphaOp.NPPI_OP_ALPHA_ATOP (C enu-
 merator), 38
 NppiAlphaOp.NPPI_OP_ALPHA_ATOP_PREMUL
 (C enumerator), 38
 NppiAlphaOp.NPPI_OP_ALPHA_IN (C enumera-
 tor), 38
 NppiAlphaOp.NPPI_OP_ALPHA_IN_PREMUL (C
 enumerator), 38
 NppiAlphaOp.NPPI_OP_ALPHA_OUT (C enumera-
 tor), 38
 NppiAlphaOp.NPPI_OP_ALPHA_OUT_PREMUL (C
 enumerator), 38
 NppiAlphaOp.NPPI_OP_ALPHA_OVER (C enu-
 merator), 38
 NppiAlphaOp.NPPI_OP_ALPHA_OVER_PREMUL
 (C enumerator), 38
 NppiAlphaOp.NPPI_OP_ALPHA_PLUS (C enu-
 merator), 38
 NppiAlphaOp.NPPI_OP_ALPHA_PLUS_PREMUL
 (C enumerator), 39
 NppiAlphaOp.NPPI_OP_ALPHA_PREMUL (C enu-
 merator), 39
 NppiAlphaOp.NPPI_OP_ALPHA_XOR (C enumer-
 ator), 38
 NppiAlphaOp.NPPI_OP_ALPHA_XOR_PREMUL (C
 enumerator), 39
 nppiAlphaPremul_8u_AC4IR_Ctx (C function),
 511
 nppiAlphaPremul_8u_AC4R_Ctx (C function),
 510
 nppiAlphaPremul_16u_AC4IR_Ctx (C func-
 tion), 511
 nppiAlphaPremul_16u_AC4R_Ctx (C function),
 511
 nppiAlphaPremulC_8u_AC4IR_Ctx (C func-
 tion), 501
 nppiAlphaPremulC_8u_AC4R_Ctx (C function),
 501
 nppiAlphaPremulC_8u_C1IR_Ctx (C function),
 499
 nppiAlphaPremulC_8u_C1R_Ctx (C function),
 499
 nppiAlphaPremulC_8u_C3IR_Ctx (C function),
 500
 nppiAlphaPremulC_8u_C3R_Ctx (C function),
 499
 nppiAlphaPremulC_8u_C4IR_Ctx (C function),
 500
 nppiAlphaPremulC_8u_C4R_Ctx (C function),
 500
 nppiAlphaPremulC_16u_AC4IR_Ctx (C func-
 tion), 504
 nppiAlphaPremulC_16u_AC4R_Ctx (C func-
 tion), 504
 nppiAlphaPremulC_16u_C1IR_Ctx (C func-
 tion), 502
 nppiAlphaPremulC_16u_C1R_Ctx (C function),
 501
 nppiAlphaPremulC_16u_C3IR_Ctx (C func-
 tion), 503
 nppiAlphaPremulC_16u_C3R_Ctx (C function),
 502
 nppiAlphaPremulC_16u_C4IR_Ctx (C func-
 tion), 503
 nppiAlphaPremulC_16u_C4R_Ctx (C function),
 503
 nppiAnd_8u_AC4IR_Ctx (C function), 463
 nppiAnd_8u_AC4R_Ctx (C function), 462
 nppiAnd_8u_C1IR_Ctx (C function), 461
 nppiAnd_8u_C1R_Ctx (C function), 461
 nppiAnd_8u_C3IR_Ctx (C function), 462
 nppiAnd_8u_C3R_Ctx (C function), 462
 nppiAnd_8u_C4IR_Ctx (C function), 464
 nppiAnd_8u_C4R_Ctx (C function), 463
 nppiAnd_16u_AC4IR_Ctx (C function), 466
 nppiAnd_16u_AC4R_Ctx (C function), 465

nppiAnd_16u_C1IR_Ctx (C function), 464
 nppiAnd_16u_C1R_Ctx (C function), 464
 nppiAnd_16u_C3IR_Ctx (C function), 465
 nppiAnd_16u_C3R_Ctx (C function), 465
 nppiAnd_16u_C4IR_Ctx (C function), 467
 nppiAnd_16u_C4R_Ctx (C function), 466
 nppiAnd_32s_AC4IR_Ctx (C function), 469
 nppiAnd_32s_AC4R_Ctx (C function), 468
 nppiAnd_32s_C1IR_Ctx (C function), 467
 nppiAnd_32s_C1R_Ctx (C function), 467
 nppiAnd_32s_C3IR_Ctx (C function), 468
 nppiAnd_32s_C3R_Ctx (C function), 468
 nppiAnd_32s_C4IR_Ctx (C function), 470
 nppiAnd_32s_C4R_Ctx (C function), 469
 nppiAndC_8u_AC4IR_Ctx (C function), 414
 nppiAndC_8u_AC4R_Ctx (C function), 414
 nppiAndC_8u_C1IR_Ctx (C function), 413
 nppiAndC_8u_C1R_Ctx (C function), 413
 nppiAndC_8u_C3IR_Ctx (C function), 414
 nppiAndC_8u_C3R_Ctx (C function), 413
 nppiAndC_8u_C4IR_Ctx (C function), 415
 nppiAndC_8u_C4R_Ctx (C function), 415
 nppiAndC_16u_AC4IR_Ctx (C function), 417
 nppiAndC_16u_AC4R_Ctx (C function), 417
 nppiAndC_16u_C1IR_Ctx (C function), 416
 nppiAndC_16u_C1R_Ctx (C function), 415
 nppiAndC_16u_C3IR_Ctx (C function), 416
 nppiAndC_16u_C3R_Ctx (C function), 416
 nppiAndC_16u_C4IR_Ctx (C function), 418
 nppiAndC_16u_C4R_Ctx (C function), 417
 nppiAndC_32s_AC4IR_Ctx (C function), 420
 nppiAndC_32s_AC4R_Ctx (C function), 420
 nppiAndC_32s_C1IR_Ctx (C function), 419
 nppiAndC_32s_C1R_Ctx (C function), 418
 nppiAndC_32s_C3IR_Ctx (C function), 419
 nppiAndC_32s_C3R_Ctx (C function), 419
 nppiAndC_32s_C4IR_Ctx (C function), 421
 nppiAndC_32s_C4R_Ctx (C function), 420
 nppiAverageError_8s_C1R_Ctx (C function), 1593
 nppiAverageError_8s_C2R_Ctx (C function), 1595
 nppiAverageError_8s_C3R_Ctx (C function), 1597
 nppiAverageError_8s_C4R_Ctx (C function), 1598
 nppiAverageError_8u_C1R_Ctx (C function), 1593
 nppiAverageError_8u_C2R_Ctx (C function), 1595
 nppiAverageError_8u_C3R_Ctx (C function), 1596
 nppiAverageError_8u_C4R_Ctx (C function), 1598
 nppiAverageError_16s_C1R_Ctx (C function), 1594
 nppiAverageError_16s_C2R_Ctx (C function), 1595
 nppiAverageError_16s_C3R_Ctx (C function), 1597
 nppiAverageError_16s_C4R_Ctx (C function), 1599
 nppiAverageError_16sc_C1R_Ctx (C function), 1594
 nppiAverageError_16sc_C2R_Ctx (C function), 1595
 nppiAverageError_16sc_C3R_Ctx (C function), 1597
 nppiAverageError_16sc_C4R_Ctx (C function), 1599
 nppiAverageError_16u_C1R_Ctx (C function), 1593
 nppiAverageError_16u_C2R_Ctx (C function), 1595
 nppiAverageError_16u_C3R_Ctx (C function), 1597
 nppiAverageError_16u_C4R_Ctx (C function), 1598
 nppiAverageError_32f_C1R_Ctx (C function), 1594
 nppiAverageError_32f_C2R_Ctx (C function), 1596
 nppiAverageError_32f_C3R_Ctx (C function), 1598
 nppiAverageError_32f_C4R_Ctx (C function), 1599
 nppiAverageError_32fc_C1R_Ctx (C function), 1594
 nppiAverageError_32fc_C2R_Ctx (C function), 1596
 nppiAverageError_32fc_C3R_Ctx (C function), 1598
 nppiAverageError_32fc_C4R_Ctx (C function), 1600
 nppiAverageError_32s_C1R_Ctx (C function), 1594
 nppiAverageError_32s_C2R_Ctx (C function), 1596
 nppiAverageError_32s_C3R_Ctx (C function), 1597
 nppiAverageError_32s_C4R_Ctx (C function), 1599
 nppiAverageError_32sc_C1R_Ctx (C function), 1594
 nppiAverageError_32sc_C2R_Ctx (C function), 1596
 nppiAverageError_32sc_C3R_Ctx (C function), 1598

nppiAverageError_32sc_C4R_Ctx (C function), 1599
 nppiAverageError_32u_C1R_Ctx (C function), 1594
 nppiAverageError_32u_C2R_Ctx (C function), 1596
 nppiAverageError_32u_C3R_Ctx (C function), 1597
 nppiAverageError_32u_C4R_Ctx (C function), 1599
 nppiAverageError_64f_C1R_Ctx (C function), 1595
 nppiAverageError_64f_C2R_Ctx (C function), 1596
 nppiAverageError_64f_C3R_Ctx (C function), 1598
 nppiAverageError_64f_C4R_Ctx (C function), 1600
 nppiAverageErrorGetBufferHostSize_8s_C1R_Ctx (C function), 1600
 nppiAverageErrorGetBufferHostSize_8s_C2R_Ctx (C function), 1602
 nppiAverageErrorGetBufferHostSize_8s_C3R_Ctx (C function), 1603
 nppiAverageErrorGetBufferHostSize_8s_C4R_Ctx (C function), 1605
 nppiAverageErrorGetBufferHostSize_8u_C1R_Ctx (C function), 1600
 nppiAverageErrorGetBufferHostSize_8u_C2R_Ctx (C function), 1602
 nppiAverageErrorGetBufferHostSize_8u_C3R_Ctx (C function), 1603
 nppiAverageErrorGetBufferHostSize_8u_C4R_Ctx (C function), 1605
 nppiAverageErrorGetBufferHostSize_16s_C1R_Ctx (C function), 1601
 nppiAverageErrorGetBufferHostSize_16s_C2R_Ctx (C function), 1602
 nppiAverageErrorGetBufferHostSize_16s_C3R_Ctx (C function), 1604
 nppiAverageErrorGetBufferHostSize_16s_C4R_Ctx (C function), 1605
 nppiAverageErrorGetBufferHostSize_16sc_C1R_Ctx (C function), 1601
 nppiAverageErrorGetBufferHostSize_16sc_C2R_Ctx (C function), 1602
 nppiAverageErrorGetBufferHostSize_16sc_C3R_Ctx (C function), 1604
 nppiAverageErrorGetBufferHostSize_16sc_C4R_Ctx (C function), 1605
 nppiAverageErrorGetBufferHostSize_16u_C1R_Ctx (C function), 1601
 nppiAverageErrorGetBufferHostSize_16u_C2R_Ctx (C function), 1602
 nppiAverageErrorGetBufferHostSize_16u_C3R_Ctx (C function), 1604
 nppiAverageErrorGetBufferHostSize_16u_C4R_Ctx (C function), 1605
 nppiAverageErrorGetBufferHostSize_32f_C1R_Ctx (C function), 1601
 nppiAverageErrorGetBufferHostSize_32f_C2R_Ctx (C function), 1603
 nppiAverageErrorGetBufferHostSize_32f_C3R_Ctx (C function), 1604
 nppiAverageErrorGetBufferHostSize_32f_C4R_Ctx (C function), 1606
 nppiAverageErrorGetBufferHostSize_32fc_C1R_Ctx (C function), 1602
 nppiAverageErrorGetBufferHostSize_32fc_C2R_Ctx (C function), 1603
 nppiAverageErrorGetBufferHostSize_32fc_C3R_Ctx (C function), 1604
 nppiAverageErrorGetBufferHostSize_32fc_C4R_Ctx (C function), 1606
 nppiAverageErrorGetBufferHostSize_32s_C1R_Ctx (C function), 1601
 nppiAverageErrorGetBufferHostSize_32s_C2R_Ctx (C function), 1603
 nppiAverageErrorGetBufferHostSize_32s_C3R_Ctx (C function), 1604
 nppiAverageErrorGetBufferHostSize_32s_C4R_Ctx (C function), 1606
 nppiAverageErrorGetBufferHostSize_32sc_C1R_Ctx (C function), 1601
 nppiAverageErrorGetBufferHostSize_32sc_C2R_Ctx (C function), 1603
 nppiAverageErrorGetBufferHostSize_32sc_C3R_Ctx (C function), 1604
 nppiAverageErrorGetBufferHostSize_32sc_C4R_Ctx (C function), 1606
 nppiAverageErrorGetBufferHostSize_32u_C1R_Ctx (C function), 1601
 nppiAverageErrorGetBufferHostSize_32u_C2R_Ctx (C function), 1602
 nppiAverageErrorGetBufferHostSize_32u_C3R_Ctx (C function), 1604
 nppiAverageErrorGetBufferHostSize_32u_C4R_Ctx (C function), 1605
 nppiAverageErrorRelativeError_8s_C1R_Ctx (C function), 1623

nppiAverageRelativeError_8s_C2R_Ctx (C function), 1625	nppiAverageRelativeError_32s_C1R_Ctx (C function), 1624
nppiAverageRelativeError_8s_C3R_Ctx (C function), 1627	nppiAverageRelativeError_32s_C2R_Ctx (C function), 1626
nppiAverageRelativeError_8s_C4R_Ctx (C function), 1629	nppiAverageRelativeError_32s_C3R_Ctx (C function), 1628
nppiAverageRelativeError_8u_C1R_Ctx (C function), 1623	nppiAverageRelativeError_32s_C4R_Ctx (C function), 1630
nppiAverageRelativeError_8u_C2R_Ctx (C function), 1625	nppiAverageRelativeError_32sc_C1R_Ctx (C function), 1624
nppiAverageRelativeError_8u_C3R_Ctx (C function), 1627	nppiAverageRelativeError_32sc_C2R_Ctx (C function), 1626
nppiAverageRelativeError_8u_C4R_Ctx (C function), 1629	nppiAverageRelativeError_32sc_C3R_Ctx (C function), 1628
nppiAverageRelativeError_16s_C1R_Ctx (C function), 1624	nppiAverageRelativeError_32sc_C4R_Ctx (C function), 1630
nppiAverageRelativeError_16s_C2R_Ctx (C function), 1626	nppiAverageRelativeError_32u_C1R_Ctx (C function), 1624
nppiAverageRelativeError_16s_C3R_Ctx (C function), 1628	nppiAverageRelativeError_32u_C2R_Ctx (C function), 1626
nppiAverageRelativeError_16s_C4R_Ctx (C function), 1629	nppiAverageRelativeError_32u_C3R_Ctx (C function), 1628
nppiAverageRelativeError_16sc_C1R_Ctx (C function), 1624	nppiAverageRelativeError_32u_C4R_Ctx (C function), 1630
nppiAverageRelativeError_16sc_C2R_Ctx (C function), 1626	nppiAverageRelativeError_64f_C1R_Ctx (C function), 1625
nppiAverageRelativeError_16sc_C3R_Ctx (C function), 1628	nppiAverageRelativeError_64f_C2R_Ctx (C function), 1627
nppiAverageRelativeError_16sc_C4R_Ctx (C function), 1630	nppiAverageRelativeError_64f_C3R_Ctx (C function), 1629
nppiAverageRelativeError_16u_C1R_Ctx (C function), 1623	nppiAverageRelativeError_64f_C4R_Ctx (C function), 1631
nppiAverageRelativeError_16u_C2R_Ctx (C function), 1625	nppiAverageRelativeErrorGetBufferHostSize_8s_C1R_Ctx (C function), 1631
nppiAverageRelativeError_16u_C3R_Ctx (C function), 1627	nppiAverageRelativeErrorGetBufferHostSize_8s_C2R_Ctx (C function), 1633
nppiAverageRelativeError_16u_C4R_Ctx (C function), 1629	nppiAverageRelativeErrorGetBufferHostSize_8s_C3R_Ctx (C function), 1635
nppiAverageRelativeError_32f_C1R_Ctx (C function), 1625	nppiAverageRelativeErrorGetBufferHostSize_8s_C4R_Ctx (C function), 1637
nppiAverageRelativeError_32f_C2R_Ctx (C function), 1626	nppiAverageRelativeErrorGetBufferHostSize_8u_C1R_Ctx (C function), 1631
nppiAverageRelativeError_32f_C3R_Ctx (C function), 1628	nppiAverageRelativeErrorGetBufferHostSize_8u_C2R_Ctx (C function), 1633
nppiAverageRelativeError_32f_C4R_Ctx (C function), 1630	nppiAverageRelativeErrorGetBufferHostSize_8u_C3R_Ctx (C function), 1635
nppiAverageRelativeError_32fc_C1R_Ctx (C function), 1625	nppiAverageRelativeErrorGetBufferHostSize_8u_C4R_Ctx (C function), 1637
nppiAverageRelativeError_32fc_C2R_Ctx (C function), 1627	nppiAverageRelativeErrorGetBufferHostSize_16s_C1R_Ctx (C function), 1632
nppiAverageRelativeError_32fc_C3R_Ctx (C function), 1629	nppiAverageRelativeErrorGetBufferHostSize_16s_C2R_Ctx (C function), 1634
nppiAverageRelativeError_32fc_C4R_Ctx (C function), 1631	nppiAverageRelativeErrorGetBufferHostSize_16s_C3R_Ctx (C function), 1636

npapiAverageRelativeErrorGetBufferHostSize_32u_C3R_Ctx (C function), 1638	npapiAverageRelativeErrorGetBufferHostSize_32u_C3R_Ctx (C function), 1636
npapiAverageRelativeErrorGetBufferHostSize_32u_C4R_Ctx (C function), 1632	npapiAverageRelativeErrorGetBufferHostSize_32u_C4R_Ctx (C function), 1638
npapiAverageRelativeErrorGetBufferHostSize_64f_C1R_Ctx (C function), 1634	npapiAverageRelativeErrorGetBufferHostSize_64f_C1R_Ctx (C function), 1633
npapiAverageRelativeErrorGetBufferHostSize_64f_C2R_Ctx (C function), 1636	npapiAverageRelativeErrorGetBufferHostSize_64f_C2R_Ctx (C function), 1635
npapiAverageRelativeErrorGetBufferHostSize_64f_C3R_Ctx (C function), 1638	npapiAverageRelativeErrorGetBufferHostSize_64f_C3R_Ctx (C function), 1637
npapiAverageRelativeErrorGetBufferHostSize_64f_C4R_Ctx (C function), 1632	npapiAverageRelativeErrorGetBufferHostSize_64f_C4R_Ctx (C function), 1639
npapiAverageRelativeErrorGetBufferHostSize_NppiAxis_Ctx (C function), 1634	NppiAxis.NPP_BOTH_AXIS (C enumerator), 35
npapiAverageRelativeErrorGetBufferHostSize_NppiAxis_HORIZONTAL_AXIS (C function), 1636	NppiAxis.NPP_HORIZONTAL_AXIS (C enumerator), 35
npapiAverageRelativeErrorGetBufferHostSize_NppiAxis_VERTICAL_AXIS (C function), 1638	NppiAxis.NPP_VERTICAL_AXIS (C enumerator), 35
npapiAverageRelativeErrorGetBufferHostSize_NppiBayerGridPosition (C function), 1633	NppiBayerGridPosition (C enum), 21
npapiAverageRelativeErrorGetBufferHostSize_32f_C2R_Ctx (C function), 1635	NppiBayerGridPosition.NPPI_BAYER_BGGR (C enumerator), 22
npapiAverageRelativeErrorGetBufferHostSize_32f_C3R_Ctx (C function), 1637	NppiBayerGridPosition.NPPI_BAYER_GBRG (C enumerator), 22
npapiAverageRelativeErrorGetBufferHostSize_32f_C4R_Ctx (C function), 1639	NppiBayerGridPosition.NPPI_BAYER_GRBG (C enumerator), 22
npapiAverageRelativeErrorGetBufferHostSize_32fc_C1R_Ctx (C function), 1633	NppiBayerGridPosition.NPPI_BAYER_RGBG (C enumerator), 22
npapiAverageRelativeErrorGetBufferHostSize_32fc_C2R_Ctx (C function), 1635	npapiBGRToCbYCr422_8u_AC4C2R_Ctx (C function), 748
npapiAverageRelativeErrorGetBufferHostSize_32fc_C3R_Ctx (C function), 1637	npapiBGRToCbYCr422_709HDTV_8u_AC4C2R_Ctx (C function), 748
npapiAverageRelativeErrorGetBufferHostSize_32fc_C4R_Ctx (C function), 1639	npapiBGRToCbYCr422_709HDTV_8u_C3C2R_Ctx (C function), 748
npapiAverageRelativeErrorGetBufferHostSize_32s_C1R_Ctx (C function), 1632	npapiBGRToHLS_8u_AC4P4R_Ctx (C function), 785
npapiAverageRelativeErrorGetBufferHostSize_32s_C2R_Ctx (C function), 1634	npapiBGRToHLS_8u_AC4R_Ctx (C function), 784
npapiAverageRelativeErrorGetBufferHostSize_32s_C3R_Ctx (C function), 1636	npapiBGRToHLS_8u_AP4C4R_Ctx (C function), 786
npapiAverageRelativeErrorGetBufferHostSize_32s_C4R_Ctx (C function), 1638	npapiBGRToHLS_8u_AP4R_Ctx (C function), 786
npapiAverageRelativeErrorGetBufferHostSize_32s_C5R_Ctx (C function), 1633	npapiBGRToHLS_8u_C3P3R_Ctx (C function), 784
npapiAverageRelativeErrorGetBufferHostSize_32sc_C1R_Ctx (C function), 1634	npapiBGRToHLS_8u_P3C3R_Ctx (C function), 785
npapiAverageRelativeErrorGetBufferHostSize_32sc_C2R_Ctx (C function), 1636	npapiBGRToHLS_8u_P3R_Ctx (C function), 786
npapiAverageRelativeErrorGetBufferHostSize_32sc_C3R_Ctx (C function), 1638	npapiBGRToHLS_8u_C3R_Ctx (C function), 774
npapiAverageRelativeErrorGetBufferHostSize_32sc_C4R_Ctx (C function), 1632	npapiBGRToYCbCr411_8u_AC4P3R_Ctx (C function), 801
npapiAverageRelativeErrorGetBufferHostSize_32u_C1R_Ctx (C function), 1632	npapiBGRToYCbCr411_8u_C3P3R_Ctx (C function), 801
npapiAverageRelativeErrorGetBufferHostSize_32u_C2R_Ctx (C function), 1634	npapiBGRToYCbCr411_JPEG_8u_C3P3R_Ctx (C function), 801
npapiAverageRelativeErrorGetBufferHostSize_32u_C3R_Ctx (C function), 1636	npapiBGRToYCbCr411_JPEG_8u_P3R_Ctx (C function), 801
npapiAverageRelativeErrorGetBufferHostSize_32u_C4R_Ctx (C function), 1638	npapiBGRToYCbCr420_8u_AC4P3R_Ctx (C function), 795
npapiAverageRelativeErrorGetBufferHostSize_32u_C5R_Ctx (C function), 1634	npapiBGRToYCbCr420_8u_C3P3R_Ctx (C function), 795

tion), 755
 nppiBGRToYCbCr420_709CSC_8u_AC4P3R_Ctx
 (C function), 756
 nppiBGRToYCbCr420_709CSC_8u_C3P3R_Ctx
 (C function), 756
 nppiBGRToYCbCr420_709HDTV_8u_AC4P3R_Ctx
 (C function), 757
 nppiBGRToYCbCr420_JPEG_8u_C3P3R_Ctx (C
 function), 800
 nppiBGRToYCbCr420_JPEG_8u_P3R_Ctx (C
 function), 794
 nppiBGRToYCbCr422_JPEG_8u_C3P3R_Ctx (C
 function), 800
 nppiBGRToYCbCr422_JPEG_8u_P3R_Ctx (C
 function), 794
 nppiBGRToYCbCr444_JPEG_8u_C3P3R_Ctx (C
 function), 801
 nppiBGRToYCbCr444_JPEG_8u_P3R_Ctx (C
 function), 795
 nppiBGRToYCbCr_8u_AC4P3R_Ctx (C function),
 764
 nppiBGRToYCbCr_8u_AC4P4R_Ctx (C function),
 764
 nppiBGRToYCbCr_8u_C3P3R_Ctx (C function),
 764
 nppiBGRToYCrCb420_8u_AC4P3R_Ctx (C func-
 tion), 762
 nppiBGRToYCrCb420_8u_C3P3R_Ctx (C func-
 tion), 762
 nppiBGRToYCrCb420_709CSC_8u_AC4P3R_Ctx
 (C function), 758
 nppiBGRToYCrCb420_709CSC_8u_C3P3R_Ctx
 (C function), 757
 nppiBGRToYUV420_8u_AC4P3R_Ctx (C func-
 tion), 720
 nppiBGRToYUV_8u_AC4P4R_Ctx (C function),
 698
 nppiBGRToYUV_8u_AC4R_Ctx (C function), 697
 nppiBGRToYUV_8u_C3P3R_Ctx (C function), 698
 nppiBGRToYUV_8u_C3R_Ctx (C function), 697
 nppiBGRToYUV_8u_P3R_Ctx (C function), 697
 NppiBorderType (C enum), 37
 NppiBorderType.NPP_BORDER_CONSTANT (C
 enumerator), 37
 NppiBorderType.NPP_BORDER_MIRROR (C enu-
 merator), 37
 NppiBorderType.NPP_BORDER_MIRROR_REFL
 (C enumerator), 37
 NppiBorderType.NPP_BORDER_NONE (C enu-
 merator), 37
 NppiBorderType.NPP_BORDER_REPLICATE (C
 enumerator), 37
 NppiBorderType.NPP_BORDER_UNDEFINED (C
 enumerator), 37
 NppiBorderType.NPP_BORDER_WRAP (C enu-
 merator), 37
 NppiBufferDescriptor (C struct), 42
 NppiBufferDescriptor.nBufferSize (C var),
 42
 NppiBufferDescriptor.pData (C var), 42
 nppiCbYCr422ToBGR_8u_C2C4R_Ctx (C func-
 tion), 749
 nppiCbYCr422ToBGR_709HDTV_8u_C2C3R_Ctx
 (C function), 750
 nppiCbYCr422ToBGR_709HDTV_8u_C2C4R_Ctx
 (C function), 750
 nppiCbYCr422ToRGB_8u_C2C3R_Ctx (C func-
 tion), 747
 nppiCbYCr422ToYCbCr411_8u_C2P3R_Ctx (C
 function), 671
 nppiCbYCr422ToYCbCr420_8u_C2P2R_Ctx (C
 function), 680
 nppiCbYCr422ToYCbCr420_8u_C2P3R_Ctx (C
 function), 680
 nppiCbYCr422ToYCbCr422_8u_C2P3R_Ctx (C
 function), 679
 nppiCbYCr422ToYCbCr422_8u_C2R_Ctx (C
 function), 679
 nppiCbYCr422ToYCrCb420_8u_C2P3R_Ctx (C
 function), 681
 nppiCFAToRGB_8u_C1C3R_Ctx (C function), 547
 nppiCFAToRGB_16u_C1C3R_Ctx (C function),
 548
 nppiCFAToRGB_32u_C1C3R_Ctx (C function),
 549
 nppiCFAToRGBA_8u_C1AC4R_Ctx (C function),
 547
 nppiCFAToRGBA_16u_C1AC4R_Ctx (C function),
 548
 nppiCFAToRGBA_32u_C1AC4R_Ctx (C function),
 549
 NppiChannels (C enum), 31
 NppiChannels.NPP_CH_1 (C enumerator), 31
 NppiChannels.NPP_CH_2 (C enumerator), 31
 NppiChannels.NPP_CH_3 (C enumerator), 32
 NppiChannels.NPP_CH_4 (C enumerator), 32
 NppiChannels.NPP_CH_A4 (C enumerator), 32
 NppiChannels.NPP_CH_P2 (C enumerator), 32
 NppiChannels.NPP_CH_P3 (C enumerator), 32
 NppiChannels.NPP_CH_P4 (C enumerator), 32
 nppiCMYK0rYCKToBGR_JPEG_8u_P4C3R_Ctx
 (C function), 780
 nppiCMYK0rYCKToBGR_JPEG_8u_P4P3R_Ctx
 (C function), 779
 nppiCMYK0rYCKToRGB_JPEG_8u_P4C3R_Ctx
 (C function), 779
 nppiCMYK0rYCKToRGB_JPEG_8u_P4P3R_Ctx
 (C function), 778

nppiColorToGray_8u_AC4C1R_Ctx (C function), 539
 nppiColorToGray_8u_C3C1R_Ctx (C function), 539
 nppiColorToGray_8u_C4C1R_Ctx (C function), 540
 nppiColorToGray_16s_AC4C1R_Ctx (C function), 542
 nppiColorToGray_16s_C3C1R_Ctx (C function), 541
 nppiColorToGray_16s_C4C1R_Ctx (C function), 542
 nppiColorToGray_16u_AC4C1R_Ctx (C function), 541
 nppiColorToGray_16u_C3C1R_Ctx (C function), 540
 nppiColorToGray_16u_C4C1R_Ctx (C function), 541
 nppiColorToGray_32f_AC4C1R_Ctx (C function), 543
 nppiColorToGray_32f_C3C1R_Ctx (C function), 543
 nppiColorToGray_32f_C4C1R_Ctx (C function), 544
 nppiColorTwist32f_8s_AC4IR_Ctx (C function), 566
 nppiColorTwist32f_8s_AC4R_Ctx (C function), 566
 nppiColorTwist32f_8s_C1IR_Ctx (C function), 563
 nppiColorTwist32f_8s_C1R_Ctx (C function), 563
 nppiColorTwist32f_8s_C2IR_Ctx (C function), 564
 nppiColorTwist32f_8s_C2R_Ctx (C function), 564
 nppiColorTwist32f_8s_C3IR_Ctx (C function), 565
 nppiColorTwist32f_8s_C3R_Ctx (C function), 564
 nppiColorTwist32f_8s_C4IR_Ctx (C function), 566
 nppiColorTwist32f_8s_C4R_Ctx (C function), 565
 nppiColorTwist32f_8s_IP3R_Ctx (C function), 567
 nppiColorTwist32f_8s_P3R_Ctx (C function), 567
 nppiColorTwist32f_8u_AC4IR_Ctx (C function), 560
 nppiColorTwist32f_8u_AC4R_Ctx (C function), 560
 nppiColorTwist32f_8u_C1IR_Ctx (C function), 557
 nppiColorTwist32f_8u_C1R_Ctx (C function), 556
 nppiColorTwist32f_8u_C2IR_Ctx (C function), 558
 nppiColorTwist32f_8u_C2R_Ctx (C function), 557
 nppiColorTwist32f_8u_C3IR_Ctx (C function), 558
 nppiColorTwist32f_8u_C3R_Ctx (C function), 558
 nppiColorTwist32f_8u_C4IR_Ctx (C function), 559
 nppiColorTwist32f_8u_C4R_Ctx (C function), 559
 nppiColorTwist32f_8u_IP3R_Ctx (C function), 562
 nppiColorTwist32f_8u_P3R_Ctx (C function), 562
 nppiColorTwist32f_16f_C1IR_Ctx (C function), 576
 nppiColorTwist32f_16f_C1R_Ctx (C function), 576
 nppiColorTwist32f_16f_C2IR_Ctx (C function), 577
 nppiColorTwist32f_16f_C2R_Ctx (C function), 576
 nppiColorTwist32f_16f_C3IR_Ctx (C function), 578
 nppiColorTwist32f_16f_C3R_Ctx (C function), 577
 nppiColorTwist32f_16f_C4IR_Ctx (C function), 579
 nppiColorTwist32f_16f_C4R_Ctx (C function), 578
 nppiColorTwist32f_16s_AC4IR_Ctx (C function), 574
 nppiColorTwist32f_16s_AC4R_Ctx (C function), 574
 nppiColorTwist32f_16s_C1IR_Ctx (C function), 572
 nppiColorTwist32f_16s_C1R_Ctx (C function), 572
 nppiColorTwist32f_16s_C2IR_Ctx (C function), 573
 nppiColorTwist32f_16s_C2R_Ctx (C function), 572
 nppiColorTwist32f_16s_C3IR_Ctx (C function), 574
 nppiColorTwist32f_16s_C3R_Ctx (C function), 573
 nppiColorTwist32f_16s_IP3R_Ctx (C function), 575
 nppiColorTwist32f_16s_P3R_Ctx (C function), 575

nppiColorTwist32f_16u_AC4IR_Ctx (C function), 570	nppiColorTwist_32fC_C4R_Ctx (C function), 584
nppiColorTwist32f_16u_AC4R_Ctx (C function), 570	nppiColorTwistBatch32f_8u_AC4IR_Ctx (C function), 590
nppiColorTwist32f_16u_C1IR_Ctx (C function), 568	nppiColorTwistBatch32f_8u_AC4R_Ctx (C function), 590
nppiColorTwist32f_16u_C1R_Ctx (C function), 568	nppiColorTwistBatch32f_8u_C1IR_Ctx (C function), 588
nppiColorTwist32f_16u_C2IR_Ctx (C function), 569	nppiColorTwistBatch32f_8u_C1R_Ctx (C function), 587
nppiColorTwist32f_16u_C2R_Ctx (C function), 568	nppiColorTwistBatch32f_8u_C3IR_Ctx (C function), 589
nppiColorTwist32f_16u_C3IR_Ctx (C function), 570	nppiColorTwistBatch32f_8u_C3R_Ctx (C function), 588
nppiColorTwist32f_16u_C3R_Ctx (C function), 569	nppiColorTwistBatch32f_8u_C4IR_Ctx (C function), 589
nppiColorTwist32f_16u_IP3R_Ctx (C function), 571	nppiColorTwistBatch32f_8u_C4R_Ctx (C function), 589
nppiColorTwist32f_16u_P3R_Ctx (C function), 571	nppiColorTwistBatch32f_16f_C1IR_Ctx (C function), 598
nppiColorTwist32fC_8u_C4IR_Ctx (C function), 561	nppiColorTwistBatch32f_16f_C1R_Ctx (C function), 597
nppiColorTwist32fC_8u_C4R_Ctx (C function), 561	nppiColorTwistBatch32f_16f_C3IR_Ctx (C function), 599
nppiColorTwist32fC_16f_C4IR_Ctx (C function), 580	nppiColorTwistBatch32f_16f_C3R_Ctx (C function), 598
nppiColorTwist32fC_16f_C4R_Ctx (C function), 579	nppiColorTwistBatch32f_16f_C4IR_Ctx (C function), 600
nppiColorTwist_32f_AC4IR_Ctx (C function), 584	nppiColorTwistBatch32f_16f_C4R_Ctx (C function), 599
nppiColorTwist_32f_AC4R_Ctx (C function), 584	nppiColorTwistBatch32fC_8u_C4IR_Ctx (C function), 592
nppiColorTwist_32f_C1IR_Ctx (C function), 581	nppiColorTwistBatch32fC_8u_C4R_Ctx (C function), 591
nppiColorTwist_32f_C1R_Ctx (C function), 580	nppiColorTwistBatch32fC_16f_C4IR_Ctx (C function), 601
nppiColorTwist_32f_C2IR_Ctx (C function), 582	nppiColorTwistBatch32fC_16f_C4R_Ctx (C function), 600
nppiColorTwist_32f_C2R_Ctx (C function), 581	nppiColorTwistBatch_32f_AC4IR_Ctx (C function), 595
nppiColorTwist_32f_C3IR_Ctx (C function), 582	nppiColorTwistBatch_32f_AC4R_Ctx (C function), 595
nppiColorTwist_32f_C3R_Ctx (C function), 582	nppiColorTwistBatch_32f_C1IR_Ctx (C function), 593
nppiColorTwist_32f_C4IR_Ctx (C function), 583	nppiColorTwistBatch_32f_C1R_Ctx (C function), 592
nppiColorTwist_32f_C4R_Ctx (C function), 583	nppiColorTwistBatch_32f_C3IR_Ctx (C function), 594
nppiColorTwist_32f_IP3R_Ctx (C function), 586	nppiColorTwistBatch_32f_C3R_Ctx (C function), 593
nppiColorTwist_32f_P3R_Ctx (C function), 586	nppiColorTwistBatch_32f_C4IR_Ctx (C function), 595
nppiColorTwist_32fC_C4IR_Ctx (C function), 585	nppiColorTwistBatch_32f_C4R_Ctx (C function), 594

nppiColorTwistBatch_32fC4IR_Ctx (C function), 597
 nppiColorTwistBatch_32fC4R_Ctx (C function), 596
 NppiColorTwistBatchCXR (C struct), 51
 NppiColorTwistBatchCXR.nDstStep (C var), 51
 NppiColorTwistBatchCXR.nSrcStep (C var), 51
 NppiColorTwistBatchCXR.pDst (C var), 51
 NppiColorTwistBatchCXR.pSrc (C var), 51
 NppiColorTwistBatchCXR.pTwist (C var), 51
 nppiCompare_8u_AC4R_Ctx (C function), 1703
 nppiCompare_8u_C1R_Ctx (C function), 1702
 nppiCompare_8u_C3R_Ctx (C function), 1702
 nppiCompare_8u_C4R_Ctx (C function), 1703
 nppiCompare_16s_AC4R_Ctx (C function), 1704
 nppiCompare_16s_C1R_Ctx (C function), 1704
 nppiCompare_16s_C3R_Ctx (C function), 1704
 nppiCompare_16s_C4R_Ctx (C function), 1704
 nppiCompare_16u_AC4R_Ctx (C function), 1703
 nppiCompare_16u_C1R_Ctx (C function), 1703
 nppiCompare_16u_C3R_Ctx (C function), 1703
 nppiCompare_16u_C4R_Ctx (C function), 1703
 nppiCompare_32f_AC4R_Ctx (C function), 1705
 nppiCompare_32f_C1R_Ctx (C function), 1704
 nppiCompare_32f_C3R_Ctx (C function), 1704
 nppiCompare_32f_C4R_Ctx (C function), 1705
 nppiCompareC_8u_AC4R_Ctx (C function), 1706
 nppiCompareC_8u_C1R_Ctx (C function), 1706
 nppiCompareC_8u_C3R_Ctx (C function), 1706
 nppiCompareC_8u_C4R_Ctx (C function), 1706
 nppiCompareC_16s_AC4R_Ctx (C function), 1708
 nppiCompareC_16s_C1R_Ctx (C function), 1707
 nppiCompareC_16s_C3R_Ctx (C function), 1707
 nppiCompareC_16s_C4R_Ctx (C function), 1707
 nppiCompareC_16u_AC4R_Ctx (C function), 1707
 nppiCompareC_16u_C1R_Ctx (C function), 1706
 nppiCompareC_16u_C3R_Ctx (C function), 1707
 nppiCompareC_16u_C4R_Ctx (C function), 1707
 nppiCompareC_32f_AC4R_Ctx (C function), 1708
 nppiCompareC_32f_C1R_Ctx (C function), 1708
 nppiCompareC_32f_C3R_Ctx (C function), 1708
 nppiCompareC_32f_C4R_Ctx (C function), 1708
 nppiCompareEqualEps_32f_AC4R_Ctx (C function), 1710
 nppiCompareEqualEps_32f_C1R_Ctx (C function), 1709
 nppiCompareEqualEps_32f_C3R_Ctx (C function), 1709
 nppiCompareEqualEps_32f_C4R_Ctx (C function), 1710
 nppiCompareEqualEpsC_32f_AC4R_Ctx (C function), 1712
 nppiCompareEqualEpsC_32f_C1R_Ctx (C function), 1711
 nppiCompareEqualEpsC_32f_C3R_Ctx (C function), 1711
 nppiCompareEqualEpsC_32f_C4R_Ctx (C function), 1712
 nppiCompColorKey_8u_C1R_Ctx (C function), 554
 nppiCompColorKey_8u_C3R_Ctx (C function), 554
 nppiCompColorKey_8u_C4R_Ctx (C function), 555
 NppiCompressedMarkerLabelsInfo (C struct), 42
 NppiCompressedMarkerLabelsInfo.nContourPixelCount (C var), 42
 NppiCompressedMarkerLabelsInfo.nContourPixelsFound (C var), 42
 NppiCompressedMarkerLabelsInfo.nMarkerLabelPixelCount (C var), 42
 NppiCompressedMarkerLabelsInfo.oContourFirstPixelLocation (C var), 42
 NppiCompressedMarkerLabelsInfo.oMarkerLabelBoundingBox (C var), 42
 nppiCompressMarkerLabelsGetBufferSize_32u_C1R (C function), 1214
 nppiCompressMarkerLabelsUF_32u_C1IR_Ctx (C function), 1214
 NppiConnectedRegion (C struct), 41
 NppiConnectedRegion.aSeedPixelValue (C var), 41
 NppiConnectedRegion.nConnectedPixelCount (C var), 41
 NppiConnectedRegion.oBoundingBox (C var), 41
 NppiContourBlockSegment (C struct), 42
 NppiContourBlockSegment.nContourPixelCount (C var), 43
 NppiContourBlockSegment.nContourStartingPixelOffset (C var), 43
 NppiContourBlockSegment.nMarkerLabelID (C var), 43
 NppiContourBlockSegment.nSegmentNum (C var), 43
 NppiContourPixelDirectionInfo (C struct), 45
 NppiContourPixelDirectionInfo.nConnected (C var), 45
 NppiContourPixelDirectionInfo.nContourDirectionCenter (C var), 45

NppiContourPixelDirectionInfo.nContourInteriorDirection (C var), 45
 NppiContourPixelDirectionInfo.nEast1 (C var), 45
 NppiContourPixelDirectionInfo.nGeometryInfoIsValid (C var), 45
 NppiContourPixelDirectionInfo.nMarkerLabelID (C var), 45
 NppiContourPixelDirectionInfo.nNorth1 (C var), 46
 NppiContourPixelDirectionInfo.nNorthEast1 (C var), 45
 NppiContourPixelDirectionInfo.nNorthWest1 (C var), 46
 NppiContourPixelDirectionInfo.nSouth1 (C var), 46
 NppiContourPixelDirectionInfo.nSouthEast1 (C var), 46
 NppiContourPixelDirectionInfo.nSouthWest1 (C var), 46
 NppiContourPixelDirectionInfo.nTest1East1 (C var), 46
 NppiContourPixelDirectionInfo.nTest1North1 (C var), 46
 NppiContourPixelDirectionInfo.nTest1NorthEast1 (C var), 46
 NppiContourPixelDirectionInfo.nTest1NorthWest1 (C var), 46
 NppiContourPixelDirectionInfo.nTest1South1 (C var), 46
 NppiContourPixelDirectionInfo.nTest1SouthEast1 (C var), 46
 NppiContourPixelDirectionInfo.nTest1SouthWest1 (C var), 46
 NppiContourPixelDirectionInfo.nTest1West1 (C var), 46
 NppiContourPixelDirectionInfo.nWest1 (C var), 46
 NppiContourPixelDirectionInfo.oContourPixelGeometryInfo (C var), 45
 NppiContourPixelGeometryInfo (C struct), 43
 NppiContourPixelGeometryInfo.nAlreadyLinked (C var), 44
 NppiContourPixelGeometryInfo.nAlreadyOutput (C var), 44
 NppiContourPixelGeometryInfo.nContourInteriorDirection (C var), 44
 NppiContourPixelGeometryInfo.nFirstIndex (C var), 43
 NppiContourPixelGeometryInfo.nLastIndex (C var), 43
 NppiContourPixelGeometryInfo.nNextContourPixelIndex (C var), 44
 NppiContourPixelGeometryInfo.nOrderIndex (C var), 44
 NppiContourPixelGeometryInfo.nPixelAlreadyUsed (C var), 44
 NppiContourPixelGeometryInfo.nPrevContourPixelIndex (C var), 44
 NppiContourPixelGeometryInfo.nReverseOrderIndex (C var), 43
 NppiContourPixelGeometryInfo.oContourCenterPixelLocation (C var), 43
 NppiContourPixelGeometryInfo.oContourNextPixelLocation (C var), 43
 NppiContourPixelGeometryInfo.oContourOrderedGeometry (C var), 43
 NppiContourPixelGeometryInfo.oContourPrevPixelLocation (C var), 43
 nppiContoursImageMarchingSquaresInterpolation_32f_Ctx (C function), 1215
 nppiContoursImageMarchingSquaresInterpolation_64f_Ctx (C function), 1218
 NppiContourTotalsInfo (C struct), 46
 NppiContourTotalsInfo.nLongestImageContourPixelCount (C var), 47
 NppiContourTotalsInfo.nTotalImagePixelContourCount (C var), 47
 nppiConnect_8u_C1Rs_Ctx (C function), 858
 nppiConvert_8s16s_C1R_Ctx (C function), 859
 nppiConnect_8u16u_C1Rs_Ctx (C function), 858
 nppiConvert_8s32f_AC4R_Ctx (C function), 855
 nppiConnect_8u32f_C1R_Ctx (C function), 855
 nppiConvert_8s32f_C3R_Ctx (C function), 855
 nppiConnect_8u32f_C4R_Ctx (C function), 855
 nppiConvert_8s32s_AC4R_Ctx (C function), 855
 nppiConnect_8u32s_C1R_Ctx (C function), 854
 nppiConvert_8s32s_C3R_Ctx (C function), 855
 nppiConvert_8s32s_C4R_Ctx (C function), 855
 nppiConnect_8u32u_C1Rs_Ctx (C function), 859
 nppiConvert_8u8s_C1RSfs_Ctx (C function), 863
 nppiConvert_8u16s_AC4R_Ctx (C function), 853
 nppiConvert_8u16s_C1R_Ctx (C function), 853
 nppiConvert_8u16s_C3R_Ctx (C function), 853
 nppiConvert_8u16s_C4R_Ctx (C function), 853
 nppiConvert_8u16u_AC4R_Ctx (C function), 853
 nppiConvert_8u16u_C1R_Ctx (C function), 852
 nppiConvert_8u16u_C3R_Ctx (C function), 852
 nppiConvert_8u16u_C4R_Ctx (C function), 852
 nppiConvert_8u32f_AC4R_Ctx (C function), 854

nppiConvert_8u32f_C1R_Ctx (C function), 854
 nppiConvert_8u32f_C3R_Ctx (C function), 854
 nppiConvert_8u32f_C4R_Ctx (C function), 854
 nppiConvert_8u32s_AC4R_Ctx (C function), 854
 nppiConvert_8u32s_C1R_Ctx (C function), 853
 nppiConvert_8u32s_C3R_Ctx (C function), 853
 nppiConvert_8u32s_C4R_Ctx (C function), 854
 nppiConvert_16f32f_AC4R_Ctx (C function), 858
 nppiConvert_16f32f_C1R_Ctx (C function), 858
 nppiConvert_16f32f_C3R_Ctx (C function), 858
 nppiConvert_16f32f_C4R_Ctx (C function), 858
 nppiConvert_16s8s_C1RSfs_Ctx (C function), 863
 nppiConvert_16s8u_AC4R_Ctx (C function), 862
 nppiConvert_16s8u_C1R_Ctx (C function), 861
 nppiConvert_16s8u_C3R_Ctx (C function), 861
 nppiConvert_16s8u_C4R_Ctx (C function), 861
 nppiConvert_16s16u_C1Rs_Ctx (C function), 859
 nppiConvert_16s32f_AC4R_Ctx (C function), 857
 nppiConvert_16s32f_C1R_Ctx (C function), 857
 nppiConvert_16s32f_C3R_Ctx (C function), 857
 nppiConvert_16s32f_C4R_Ctx (C function), 857
 nppiConvert_16s32s_AC4R_Ctx (C function), 857
 nppiConvert_16s32s_C1R_Ctx (C function), 857
 nppiConvert_16s32s_C3R_Ctx (C function), 857
 nppiConvert_16s32s_C4R_Ctx (C function), 857
 nppiConvert_16s32u_C1Rs_Ctx (C function), 859
 nppiConvert_16u8s_C1RSfs_Ctx (C function), 863
 nppiConvert_16u8u_AC4R_Ctx (C function), 861
 nppiConvert_16u8u_C1R_Ctx (C function), 861
 nppiConvert_16u8u_C3R_Ctx (C function), 861
 nppiConvert_16u8u_C4R_Ctx (C function), 861
 nppiConvert_16u16s_C1RSfs_Ctx (C function), 863
 nppiConvert_16u32f_AC4R_Ctx (C function), 856
 nppiConvert_16u32f_C1R_Ctx (C function), 856
 nppiConvert_16u32f_C3R_Ctx (C function), 856
 nppiConvert_16u32f_C4R_Ctx (C function), 856
 nppiConvert_16u32s_AC4R_Ctx (C function), 856
 nppiConvert_16u32s_C1R_Ctx (C function), 855
 nppiConvert_16u32s_C3R_Ctx (C function), 856
 nppiConvert_16u32s_C4R_Ctx (C function), 856
 nppiConvert_16u32u_C1R_Ctx (C function), 859
 nppiConvert_32f8s_AC4R_Ctx (C function), 865
 nppiConvert_32f8s_C1R_Ctx (C function), 865
 nppiConvert_32f8s_C1RSfs_Ctx (C function), 868
 nppiConvert_32f8s_C3R_Ctx (C function), 865
 nppiConvert_32f8s_C4R_Ctx (C function), 865
 nppiConvert_32f8u_AC4R_Ctx (C function), 865
 nppiConvert_32f8u_C1R_Ctx (C function), 865
 nppiConvert_32f8u_C1RSfs_Ctx (C function), 867
 nppiConvert_32f8u_C3R_Ctx (C function), 865
 nppiConvert_32f8u_C4R_Ctx (C function), 865
 nppiConvert_32f16f_AC4R_Ctx (C function), 867
 nppiConvert_32f16f_C1R_Ctx (C function), 867
 nppiConvert_32f16f_C3R_Ctx (C function), 867
 nppiConvert_32f16f_C4R_Ctx (C function), 867
 nppiConvert_32f16s_AC4R_Ctx (C function), 867
 nppiConvert_32f16s_C1R_Ctx (C function), 866
 nppiConvert_32f16s_C1RSfs_Ctx (C function), 868
 nppiConvert_32f16s_C3R_Ctx (C function), 866
 nppiConvert_32f16s_C4R_Ctx (C function), 866
 nppiConvert_32f16u_AC4R_Ctx (C function), 866
 nppiConvert_32f16u_C1R_Ctx (C function), 866
 nppiConvert_32f16u_C1RSfs_Ctx (C function), 868

nppiConvert_32f16u_C3R_Ctx (C function), 866
 nppiConvert_32f16u_C4R_Ctx (C function), 866
 nppiConvert_32f32s_C1RSfs_Ctx (C function), 868
 nppiConvert_32f32u_C1RSfs_Ctx (C function), 868
 nppiConvert_32s8s_AC4R_Ctx (C function), 863
 nppiConvert_32s8s_C1R_Ctx (C function), 862
 nppiConvert_32s8s_C3R_Ctx (C function), 862
 nppiConvert_32s8s_C4R_Ctx (C function), 863
 nppiConvert_32s8u_AC4R_Ctx (C function), 862
 nppiConvert_32s8u_C1R_Ctx (C function), 862
 nppiConvert_32s8u_C3R_Ctx (C function), 862
 nppiConvert_32s8u_C4R_Ctx (C function), 862
 nppiConvert_32s16s_C1RSfs_Ctx (C function), 864
 nppiConvert_32s16u_C1RSfs_Ctx (C function), 864
 nppiConvert_32s32f_C1R_Ctx (C function), 859
 nppiConvert_32s32u_C1Rs_Ctx (C function), 859
 nppiConvert_32u8s_C1RSfs_Ctx (C function), 864
 nppiConvert_32u8u_C1RSfs_Ctx (C function), 863
 nppiConvert_32u16s_C1RSfs_Ctx (C function), 864
 nppiConvert_32u16u_C1RSfs_Ctx (C function), 864
 nppiConvert_32u32f_C1R_Ctx (C function), 860
 nppiConvert_32u32s_C1RSfs_Ctx (C function), 864
 nppiCopy_8s_AC4R_Ctx (C function), 817
 nppiCopy_8s_C1R_Ctx (C function), 817
 nppiCopy_8s_C2R_Ctx (C function), 817
 nppiCopy_8s_C3R_Ctx (C function), 817
 nppiCopy_8s_C4R_Ctx (C function), 817
 nppiCopy_8u_AC4MR_Ctx (C function), 822
 nppiCopy_8u_AC4R_Ctx (C function), 817
 nppiCopy_8u_C1C3R_Ctx (C function), 829
 nppiCopy_8u_C1C4R_Ctx (C function), 829
 nppiCopy_8u_C1MR_Ctx (C function), 822
 nppiCopy_8u_C1R_Ctx (C function), 817
 nppiCopy_8u_C3C1R_Ctx (C function), 827
 nppiCopy_8u_C3CR_Ctx (C function), 825
 nppiCopy_8u_C3MR_Ctx (C function), 822
 nppiCopy_8u_C3P3R_Ctx (C function), 831
 nppiCopy_8u_C3R_Ctx (C function), 817
 nppiCopy_8u_C4C1R_Ctx (C function), 827
 nppiCopy_8u_C4CR_Ctx (C function), 825
 nppiCopy_8u_C4MR_Ctx (C function), 822
 nppiCopy_8u_C4P4R_Ctx (C function), 831
 nppiCopy_8u_C4R_Ctx (C function), 817
 nppiCopy_8u_P3C3R_Ctx (C function), 833
 nppiCopy_8u_P4C4R_Ctx (C function), 833
 nppiCopy_16f_C1R_Ctx (C function), 820
 nppiCopy_16f_C3R_Ctx (C function), 820
 nppiCopy_16f_C4R_Ctx (C function), 820
 nppiCopy_16s_AC4MR_Ctx (C function), 823
 nppiCopy_16s_AC4R_Ctx (C function), 818
 nppiCopy_16s_C1C3R_Ctx (C function), 829
 nppiCopy_16s_C1C4R_Ctx (C function), 829
 nppiCopy_16s_C1MR_Ctx (C function), 823
 nppiCopy_16s_C1R_Ctx (C function), 818
 nppiCopy_16s_C3C1R_Ctx (C function), 827
 nppiCopy_16s_C3CR_Ctx (C function), 825
 nppiCopy_16s_C3MR_Ctx (C function), 823
 nppiCopy_16s_C3P3R_Ctx (C function), 831
 nppiCopy_16s_C3R_Ctx (C function), 818
 nppiCopy_16s_C4C1R_Ctx (C function), 827
 nppiCopy_16s_C4CR_Ctx (C function), 825
 nppiCopy_16s_C4MR_Ctx (C function), 823
 nppiCopy_16s_C4P4R_Ctx (C function), 831
 nppiCopy_16s_C4R_Ctx (C function), 818
 nppiCopy_16s_P3C3R_Ctx (C function), 833
 nppiCopy_16s_P4C4R_Ctx (C function), 834
 nppiCopy_16sc_AC4R_Ctx (C function), 819
 nppiCopy_16sc_C1R_Ctx (C function), 818
 nppiCopy_16sc_C2R_Ctx (C function), 818
 nppiCopy_16sc_C3R_Ctx (C function), 818
 nppiCopy_16sc_C4R_Ctx (C function), 819
 nppiCopy_16u_AC4MR_Ctx (C function), 823
 nppiCopy_16u_AC4R_Ctx (C function), 818
 nppiCopy_16u_C1C3R_Ctx (C function), 829
 nppiCopy_16u_C1C4R_Ctx (C function), 829
 nppiCopy_16u_C1MR_Ctx (C function), 823
 nppiCopy_16u_C1R_Ctx (C function), 817
 nppiCopy_16u_C3C1R_Ctx (C function), 827
 nppiCopy_16u_C3CR_Ctx (C function), 826
 nppiCopy_16u_C3MR_Ctx (C function), 823
 nppiCopy_16u_C3P3R_Ctx (C function), 831
 nppiCopy_16u_C3R_Ctx (C function), 818
 nppiCopy_16u_C4C1R_Ctx (C function), 827
 nppiCopy_16u_C4CR_Ctx (C function), 826
 nppiCopy_16u_C4MR_Ctx (C function), 823
 nppiCopy_16u_C4P4R_Ctx (C function), 832
 nppiCopy_16u_C4R_Ctx (C function), 818
 nppiCopy_16u_P3C3R_Ctx (C function), 833
 nppiCopy_16u_P4C4R_Ctx (C function), 833
 nppiCopy_32f_AC4MR_Ctx (C function), 824
 nppiCopy_32f_AC4R_Ctx (C function), 820
 nppiCopy_32f_C1C2R_Ctx (C function), 830

nppiCopy_32f_C1C3R_Ctx (C function), 830
 nppiCopy_32f_C1C4R_Ctx (C function), 830
 nppiCopy_32f_C1MR_Ctx (C function), 824
 nppiCopy_32f_C1R_Ctx (C function), 820
 nppiCopy_32f_C2C1R_Ctx (C function), 828
 nppiCopy_32f_C3C1R_Ctx (C function), 828
 nppiCopy_32f_C3CR_Ctx (C function), 826
 nppiCopy_32f_C3MR_Ctx (C function), 824
 nppiCopy_32f_C3P3R_Ctx (C function), 832
 nppiCopy_32f_C3R_Ctx (C function), 820
 nppiCopy_32f_C4C1R_Ctx (C function), 828
 nppiCopy_32f_C4CR_Ctx (C function), 826
 nppiCopy_32f_C4MR_Ctx (C function), 824
 nppiCopy_32f_C4P4R_Ctx (C function), 832
 nppiCopy_32f_C4R_Ctx (C function), 820
 nppiCopy_32f_P3C3R_Ctx (C function), 834
 nppiCopy_32f_P4C4R_Ctx (C function), 834
 nppiCopy_32fc_AC4R_Ctx (C function), 821
 nppiCopy_32fc_C1R_Ctx (C function), 821
 nppiCopy_32fc_C2R_Ctx (C function), 821
 nppiCopy_32fc_C3R_Ctx (C function), 821
 nppiCopy_32fc_C4R_Ctx (C function), 821
 nppiCopy_32s_AC4MR_Ctx (C function), 824
 nppiCopy_32s_AC4R_Ctx (C function), 819
 nppiCopy_32s_C1C3R_Ctx (C function), 830
 nppiCopy_32s_C1C4R_Ctx (C function), 830
 nppiCopy_32s_C1MR_Ctx (C function), 823
 nppiCopy_32s_C1R_Ctx (C function), 819
 nppiCopy_32s_C3C1R_Ctx (C function), 828
 nppiCopy_32s_C3CR_Ctx (C function), 826
 nppiCopy_32s_C3MR_Ctx (C function), 824
 nppiCopy_32s_C3P3R_Ctx (C function), 832
 nppiCopy_32s_C3R_Ctx (C function), 819
 nppiCopy_32s_C4C1R_Ctx (C function), 828
 nppiCopy_32s_C4CR_Ctx (C function), 826
 nppiCopy_32s_C4MR_Ctx (C function), 824
 nppiCopy_32s_C4P4R_Ctx (C function), 832
 nppiCopy_32s_C4R_Ctx (C function), 819
 nppiCopy_32s_P3C3R_Ctx (C function), 834
 nppiCopy_32s_P4C4R_Ctx (C function), 834
 nppiCopy_32sc_AC4R_Ctx (C function), 820
 nppiCopy_32sc_C1R_Ctx (C function), 819
 nppiCopy_32sc_C2R_Ctx (C function), 819
 nppiCopy_32sc_C3R_Ctx (C function), 819
 nppiCopy_32sc_C4R_Ctx (C function), 820
 nppiCopyConstBorder_8u_AC4R_Ctx (C function), 836
 nppiCopyConstBorder_8u_C1R_Ctx (C function), 835
 nppiCopyConstBorder_8u_C3R_Ctx (C function), 835
 nppiCopyConstBorder_8u_C4R_Ctx (C function), 835
 nppiCopyConstBorder_16s_AC4R_Ctx (C function), 837
 nppiCopyConstBorder_16s_C1R_Ctx (C function), 836
 nppiCopyConstBorder_16s_C3R_Ctx (C function), 837
 nppiCopyConstBorder_16s_C4R_Ctx (C function), 837
 nppiCopyConstBorder_16u_AC4R_Ctx (C function), 836
 nppiCopyConstBorder_16u_C1R_Ctx (C function), 836
 nppiCopyConstBorder_16u_C3R_Ctx (C function), 836
 nppiCopyConstBorder_16u_C4R_Ctx (C function), 836
 nppiCopyConstBorder_32f_AC4R_Ctx (C function), 838
 nppiCopyConstBorder_32f_C1R_Ctx (C function), 838
 nppiCopyConstBorder_32f_C3R_Ctx (C function), 838
 nppiCopyConstBorder_32f_C4R_Ctx (C function), 838
 nppiCopyConstBorder_32s_AC4R_Ctx (C function), 838
 nppiCopyConstBorder_32s_C1R_Ctx (C function), 837
 nppiCopyConstBorder_32s_C3R_Ctx (C function), 837
 nppiCopyConstBorder_32s_C4R_Ctx (C function), 838
 nppiCopyReplicateBorder_8u_AC4R_Ctx (C function), 840
 nppiCopyReplicateBorder_8u_C1R_Ctx (C function), 840
 nppiCopyReplicateBorder_8u_C3R_Ctx (C function), 840
 nppiCopyReplicateBorder_8u_C4R_Ctx (C function), 840
 nppiCopyReplicateBorder_16s_AC4R_Ctx (C function), 841
 nppiCopyReplicateBorder_16s_C1R_Ctx (C function), 841
 nppiCopyReplicateBorder_16s_C3R_Ctx (C function), 841
 nppiCopyReplicateBorder_16s_C4R_Ctx (C function), 841
 nppiCopyReplicateBorder_16u_AC4R_Ctx (C function), 841
 nppiCopyReplicateBorder_16u_C1R_Ctx (C function), 840
 nppiCopyReplicateBorder_16u_C3R_Ctx (C function), 840

nppiCopyReplicateBorder_16u_C4R_Ctx (C function), 841	nppiCopyWrapBorder_8u_AC4R_Ctx (C function), 845
nppiCopyReplicateBorder_32f_AC4R_Ctx (C function), 843	nppiCopyWrapBorder_8u_C1R_Ctx (C function), 844
nppiCopyReplicateBorder_32f_C1R_Ctx (C function), 842	nppiCopyWrapBorder_8u_C3R_Ctx (C function), 844
nppiCopyReplicateBorder_32f_C3R_Ctx (C function), 843	nppiCopyWrapBorder_8u_C4R_Ctx (C function), 844
nppiCopyReplicateBorder_32f_C4R_Ctx (C function), 843	nppiCopyWrapBorder_16s_AC4R_Ctx (C function), 846
nppiCopyReplicateBorder_32s_AC4R_Ctx (C function), 842	nppiCopyWrapBorder_16s_C1R_Ctx (C function), 846
nppiCopyReplicateBorder_32s_C1R_Ctx (C function), 842	nppiCopyWrapBorder_16s_C3R_Ctx (C function), 846
nppiCopyReplicateBorder_32s_C3R_Ctx (C function), 842	nppiCopyWrapBorder_16s_C4R_Ctx (C function), 846
nppiCopyReplicateBorder_32s_C4R_Ctx (C function), 842	nppiCopyWrapBorder_16u_AC4R_Ctx (C function), 845
nppiCopySubpix_8u_AC4R_Ctx (C function), 849	nppiCopyWrapBorder_16u_C1R_Ctx (C function), 845
nppiCopySubpix_8u_C1R_Ctx (C function), 849	nppiCopyWrapBorder_16u_C3R_Ctx (C function), 845
nppiCopySubpix_8u_C3R_Ctx (C function), 849	nppiCopyWrapBorder_16u_C4R_Ctx (C function), 845
nppiCopySubpix_8u_C4R_Ctx (C function), 849	nppiCopyWrapBorder_32f_AC4R_Ctx (C function), 848
nppiCopySubpix_16s_AC4R_Ctx (C function), 850	nppiCopyWrapBorder_32f_C1R_Ctx (C function), 847
nppiCopySubpix_16s_C1R_Ctx (C function), 850	nppiCopyWrapBorder_32f_C3R_Ctx (C function), 847
nppiCopySubpix_16s_C3R_Ctx (C function), 850	nppiCopyWrapBorder_32f_C4R_Ctx (C function), 847
nppiCopySubpix_16s_C4R_Ctx (C function), 850	nppiCopyWrapBorder_32s_AC4R_Ctx (C function), 847
nppiCopySubpix_16u_AC4R_Ctx (C function), 850	nppiCopyWrapBorder_32s_C1R_Ctx (C function), 846
nppiCopySubpix_16u_C1R_Ctx (C function), 849	nppiCopyWrapBorder_32s_C3R_Ctx (C function), 846
nppiCopySubpix_16u_C3R_Ctx (C function), 849	nppiCopyWrapBorder_32s_C4R_Ctx (C function), 847
nppiCopySubpix_16u_C4R_Ctx (C function), 849	nppiCountInRange_8u_AC4R_Ctx (C function), 1476
nppiCopySubpix_32f_AC4R_Ctx (C function), 851	nppiCountInRange_8u_C1R_Ctx (C function), 1475
nppiCopySubpix_32f_C1R_Ctx (C function), 851	nppiCountInRange_8u_C3R_Ctx (C function), 1475
nppiCopySubpix_32f_C3R_Ctx (C function), 851	nppiCountInRange_32f_AC4R_Ctx (C function), 1476
nppiCopySubpix_32f_C4R_Ctx (C function), 851	nppiCountInRange_32f_C1R_Ctx (C function), 1475
nppiCopySubpix_32s_AC4R_Ctx (C function), 851	nppiCountInRange_32f_C3R_Ctx (C function), 1476
nppiCopySubpix_32s_C1R_Ctx (C function), 850	nppiCountInRangeGetBufferHostSize_8u_AC4R_Ctx
nppiCopySubpix_32s_C3R_Ctx (C function), 850	
nppiCopySubpix_32s_C4R_Ctx (C function),	

(C function), 1477
 nppiCountInRangeGetBufferHostSize_8u_C1RnppCrossCorrFull_Norm_64f_C3R_Ctx (C
 (C function), 1476
 nppiCountInRangeGetBufferHostSize_8u_C3RnppCrossCorrFull_Norm_64f_C4R_Ctx (C
 (C function), 1477
 nppiCountInRangeGetBufferHostSize_32f_AC4RnppCrossCorrFull_NormLevel_8s32f_AC4R_Ctx
 (C function), 1477
 nppiCountInRangeGetBufferHostSize_32f_C1RnppCrossCorrFull_NormLevel_8s32f_C1R_Ctx
 (C function), 1477
 nppiCountInRangeGetBufferHostSize_32f_C3RnppCrossCorrFull_NormLevel_8s32f_C3R_Ctx
 (C function), 1477
 nppiCrossCorrFull_Norm_8s32f_AC4R_Ctx nppiCrossCorrFull_NormLevel_8s32f_C4R_Ctx
 (C function), 1519
 nppiCrossCorrFull_Norm_8s32f_C1R_Ctx (C nppiCrossCorrFull_NormLevel_8u32f_AC4R_Ctx
 function), 1518
 nppiCrossCorrFull_Norm_8s32f_C3R_Ctx (C nppiCrossCorrFull_NormLevel_8u32f_C1R_Ctx
 function), 1518
 nppiCrossCorrFull_Norm_8s32f_C4R_Ctx (C nppiCrossCorrFull_NormLevel_8u32f_C3R_Ctx
 function), 1518
 nppiCrossCorrFull_Norm_8u32f_AC4R_Ctx nppiCrossCorrFull_NormLevel_8u32f_C4R_Ctx
 (C function), 1518
 nppiCrossCorrFull_Norm_8u32f_C1R_Ctx (C nppiCrossCorrFull_NormLevel_8u_AC4RSfs_Ctx
 function), 1517
 nppiCrossCorrFull_Norm_8u32f_C3R_Ctx (C nppiCrossCorrFull_NormLevel_8u_C1RSfs_Ctx
 function), 1517
 nppiCrossCorrFull_Norm_8u32f_C4R_Ctx (C nppiCrossCorrFull_NormLevel_8u_C3RSfs_Ctx
 function), 1518
 nppiCrossCorrFull_Norm_8u_AC4RSfs_Ctx nppiCrossCorrFull_NormLevel_8u_C4RSfs_Ctx
 (C function), 1516
 nppiCrossCorrFull_Norm_8u_C1RSfs_Ctx (C nppiCrossCorrFull_NormLevel_16u32f_AC4R_Ctx
 function), 1515
 nppiCrossCorrFull_Norm_8u_C3RSfs_Ctx (C nppiCrossCorrFull_NormLevel_16u32f_C1R_Ctx
 function), 1515
 nppiCrossCorrFull_Norm_8u_C4RSfs_Ctx (C nppiCrossCorrFull_NormLevel_16u32f_C3R_Ctx
 function), 1515
 nppiCrossCorrFull_Norm_16u32f_AC4R_Ctx nppiCrossCorrFull_NormLevel_16u32f_C4R_Ctx
 (C function), 1519
 nppiCrossCorrFull_Norm_16u32f_C1R_Ctx nppiCrossCorrFull_NormLevel_32f_AC4R_Ctx
 (C function), 1519
 nppiCrossCorrFull_Norm_16u32f_C3R_Ctx nppiCrossCorrFull_NormLevel_32f_C1R_Ctx
 (C function), 1519
 nppiCrossCorrFull_Norm_16u32f_C4R_Ctx nppiCrossCorrFull_NormLevel_32f_C3R_Ctx
 (C function), 1519
 nppiCrossCorrFull_Norm_32f_AC4R_Ctx (C nppiCrossCorrFull_NormLevel_32f_C4R_Ctx
 function), 1516
 nppiCrossCorrFull_Norm_32f_C1R_Ctx (C nppiCrossCorrFull_NormLevel_64f_AC4R_Ctx
 function), 1516
 nppiCrossCorrFull_Norm_32f_C3R_Ctx (C nppiCrossCorrFull_NormLevel_64f_C1R_Ctx
 function), 1516
 nppiCrossCorrFull_Norm_32f_C4R_Ctx (C nppiCrossCorrFull_NormLevel_64f_C3R_Ctx
 function), 1516
 nppiCrossCorrFull_Norm_64f_AC4R_Ctx (C nppiCrossCorrFull_NormLevel_64f_C4R_Ctx
 function), 1517
 nppiCrossCorrFull_Norm_64f_C1R_Ctx (C nppiCrossCorrFull_NormLevel_GetAdvancedScratchBuff

(C function), 1562
 nppiCrossCorrFull_NormLevelAdvanced_8s32f_AC4R_Ctx (C function), 1560
 nppiCrossCorrFull_NormLevelAdvanced_8s32f_C1R_Ctx (C function), 1560
 nppiCrossCorrFull_NormLevelAdvanced_8s32f_C3R_Ctx (C function), 1561
 nppiCrossCorrFull_NormLevelAdvanced_8u32f_AC4R_Ctx (C function), 1559
 nppiCrossCorrFull_NormLevelAdvanced_8u32f_C1R_Ctx (C function), 1560
 nppiCrossCorrFull_NormLevelAdvanced_8u32f_C3R_Ctx (C function), 1561
 nppiCrossCorrFull_NormLevelAdvanced_16u32f_AC4R_Ctx (C function), 1561
 nppiCrossCorrFull_NormLevelAdvanced_16u32f_C1R_Ctx (C function), 1561
 nppiCrossCorrFull_NormLevelAdvanced_16u32f_C3R_Ctx (C function), 1562
 nppiCrossCorrFull_NormLevelAdvanced_32f_AC4R_Ctx (C function), 1558
 nppiCrossCorrFull_NormLevelAdvanced_32f_C1R_Ctx (C function), 1558
 nppiCrossCorrFull_NormLevelAdvanced_32f_C3R_Ctx (C function), 1558
 nppiCrossCorrFull_NormLevelAdvanced_64f_AC4R_Ctx (C function), 1558
 nppiCrossCorrFull_NormLevelAdvanced_64f_C1R_Ctx (C function), 1559
 nppiCrossCorrFull_NormLevelAdvanced_64f_C3R_Ctx (C function), 1559
 nppiCrossCorrSame_Norm_8s32f_AC4R_Ctx (C function), 1523
 nppiCrossCorrSame_Norm_8s32f_C1R_Ctx (C function), 1523
 nppiCrossCorrSame_Norm_8s32f_C3R_Ctx (C function), 1523
 nppiCrossCorrSame_Norm_8s32f_C4R_Ctx (C function), 1523
 nppiCrossCorrSame_Norm_8u32f_AC4R_Ctx (C function), 1522
 nppiCrossCorrSame_Norm_8u32f_C1R_Ctx (C function), 1522
 nppiCrossCorrSame_Norm_8u32f_C3R_Ctx (C function), 1522
 nppiCrossCorrSame_Norm_8u32f_C4R_Ctx (C function), 1522
 nppiCrossCorrSame_Norm_8u_AC4RSfs_Ctx (C function), 1520
 nppiCrossCorrSame_Norm_8u_C1RSfs_Ctx (C function), 1520
 nppiCrossCorrSame_Norm_8u_C3RSfs_Ctx (C function), 1520
 nppiCrossCorrSame_Norm_8u_C4RSfs_Ctx (C function), 1520
 function), 1520
 nppiCrossCorrSame_Norm_16u32f_AC4R_Ctx (C function), 1524
 nppiCrossCorrSame_Norm_16u32f_C1R_Ctx (C function), 1523
 nppiCrossCorrSame_Norm_16u32f_C3R_Ctx (C function), 1523
 nppiCrossCorrSame_Norm_16u32f_C4R_Ctx (C function), 1524
 nppiCrossCorrSame_Norm_32f_AC4R_Ctx (C function), 1521
 nppiCrossCorrSame_Norm_32f_C1R_Ctx (C function), 1520
 nppiCrossCorrSame_Norm_32f_C3R_Ctx (C function), 1521
 nppiCrossCorrSame_Norm_32f_C4R_Ctx (C function), 1521
 nppiCrossCorrSame_Norm_64f_AC4R_Ctx (C function), 1522
 nppiCrossCorrSame_Norm_64f_C1R_Ctx (C function), 1521
 nppiCrossCorrSame_Norm_64f_C3R_Ctx (C function), 1521
 nppiCrossCorrSame_Norm_64f_C4R_Ctx (C function), 1522
 nppiCrossCorrSame_NormLevel_8s32f_AC4R_Ctx (C function), 1543
 nppiCrossCorrSame_NormLevel_8s32f_C1R_Ctx (C function), 1542
 nppiCrossCorrSame_NormLevel_8s32f_C3R_Ctx (C function), 1542
 nppiCrossCorrSame_NormLevel_8s32f_C4R_Ctx (C function), 1543
 nppiCrossCorrSame_NormLevel_8u32f_AC4R_Ctx (C function), 1542
 nppiCrossCorrSame_NormLevel_8u32f_C1R_Ctx (C function), 1541
 nppiCrossCorrSame_NormLevel_8u32f_C3R_Ctx (C function), 1542
 nppiCrossCorrSame_NormLevel_8u32f_C4R_Ctx (C function), 1542
 nppiCrossCorrSame_NormLevel_8u_AC4RSfs_Ctx (C function), 1540
 nppiCrossCorrSame_NormLevel_8u_C1RSfs_Ctx (C function), 1539
 nppiCrossCorrSame_NormLevel_8u_C3RSfs_Ctx (C function), 1539
 nppiCrossCorrSame_NormLevel_8u_C4RSfs_Ctx (C function), 1539
 nppiCrossCorrSame_NormLevel_16u32f_AC4R_Ctx (C function), 1544
 nppiCrossCorrSame_NormLevel_16u32f_C1R_Ctx (C function), 1543
 nppiCrossCorrSame_NormLevel_16u32f_C3R_Ctx

(C function), 1543
 nppiCrossCorrSame_NormLevel_16u32f_C4R_Ctx (C function), 1543
 nppiCrossCorrSame_NormLevel_32f_AC4R_Ctx (C function), 1540
 nppiCrossCorrSame_NormLevel_32f_C1R_Ctx (C function), 1540
 nppiCrossCorrSame_NormLevel_32f_C3R_Ctx (C function), 1540
 nppiCrossCorrSame_NormLevel_32f_C4R_Ctx (C function), 1540
 nppiCrossCorrSame_NormLevel_64f_AC4R_Ctx (C function), 1541
 nppiCrossCorrSame_NormLevel_64f_C1R_Ctx (C function), 1541
 nppiCrossCorrSame_NormLevel_64f_C3R_Ctx (C function), 1541
 nppiCrossCorrSame_NormLevel_64f_C4R_Ctx (C function), 1541
 nppiCrossCorrSame_NormLevel_GetAdvancedSupportInfo (C function), 1568
 nppiCrossCorrSame_NormLevelAdvanced_8s32f_C1R_Ctx (C function), 1566
 nppiCrossCorrSame_NormLevelAdvanced_8s32f_C3R_Ctx (C function), 1566
 nppiCrossCorrSame_NormLevelAdvanced_8s32f_C4R_Ctx (C function), 1566
 nppiCrossCorrSame_NormLevelAdvanced_8u32f_C1R_Ctx (C function), 1565
 nppiCrossCorrSame_NormLevelAdvanced_8u32f_C3R_Ctx (C function), 1565
 nppiCrossCorrSame_NormLevelAdvanced_8u32f_C4R_Ctx (C function), 1565
 nppiCrossCorrSame_NormLevelAdvanced_16u32f_AC4R_Ctx (C function), 1567
 nppiCrossCorrSame_NormLevelAdvanced_16u32f_C1R_Ctx (C function), 1567
 nppiCrossCorrSame_NormLevelAdvanced_16u32f_C3R_Ctx (C function), 1567
 nppiCrossCorrSame_NormLevelAdvanced_16u32f_C4R_Ctx (C function), 1567
 nppiCrossCorrSame_NormLevelAdvanced_32f_AC4R_Ctx (C function), 1563
 nppiCrossCorrSame_NormLevelAdvanced_32f_C1R_Ctx (C function), 1563
 nppiCrossCorrSame_NormLevelAdvanced_32f_C3R_Ctx (C function), 1564
 nppiCrossCorrSame_NormLevelAdvanced_64f_AC4R_Ctx (C function), 1564
 nppiCrossCorrSame_NormLevelAdvanced_64f_C1R_Ctx (C function), 1564
 nppiCrossCorrValid_8s32f_C1R_Ctx (C function), 1529
 nppiCrossCorrValid_8u32f_C1R_Ctx (C function), 1529
 nppiCrossCorrValid_16u32f_C1R_Ctx (C function), 1530
 nppiCrossCorrValid_32f_C1R_Ctx (C function), 1529
 nppiCrossCorrValid_64f_C1R_Ctx (C function), 1529
 nppiCrossCorrValid_Norm_8s32f_AC4R_Ctx (C function), 1528
 nppiCrossCorrValid_Norm_8s32f_C1R_Ctx (C function), 1527
 nppiCrossCorrValid_Norm_8s32f_C3R_Ctx (C function), 1527
 nppiCrossCorrValid_Norm_8s32f_C4R_Ctx (C function), 1528
 nppiCrossCorrValid_Norm_8u32f_AC4R_Ctx (C function), 1527
 nppiCrossCorrValid_Norm_8u32f_C1R_Ctx (C function), 1527
 nppiCrossCorrValid_Norm_8u32f_C3R_Ctx (C function), 1527
 nppiCrossCorrValid_Norm_8u32f_C4R_Ctx (C function), 1527
 nppiCrossCorrValid_Norm_8u_AC4RSfs_Ctx (C function), 1525
 nppiCrossCorrValid_Norm_8u_C1RSfs_Ctx (C function), 1524
 nppiCrossCorrValid_Norm_8u_C3RSfs_Ctx (C function), 1524
 nppiCrossCorrValid_Norm_8u_C4RSfs_Ctx (C function), 1525
 nppiCrossCorrValid_Norm_16u32f_AC4R_Ctx (C function), 1529
 nppiCrossCorrValid_Norm_16u32f_C1R_Ctx (C function), 1528
 nppiCrossCorrValid_Norm_16u32f_C3R_Ctx (C function), 1528
 nppiCrossCorrValid_Norm_16u32f_C4R_Ctx (C function), 1528
 nppiCrossCorrValid_Norm_32f_AC4R_Ctx (C function), 1526
 nppiCrossCorrValid_Norm_32f_C1R_Ctx (C function), 1525
 nppiCrossCorrValid_Norm_32f_C3R_Ctx (C function), 1525
 nppiCrossCorrValid_Norm_32f_C4R_Ctx (C function), 1525
 nppiCrossCorrValid_Norm_64f_AC4R_Ctx (C function), 1526
 nppiCrossCorrValid_Norm_64f_C1R_Ctx (C function), 1526
 nppiCrossCorrValid_Norm_64f_C3R_Ctx (C function), 1526
 nppiCrossCorrValid_Norm_64f_C4R_Ctx (C function), 1526

function), 1526
 nppiCrossCorrValid_NormLevel_8s32f_AC4R_Ctx (C function), 1552
 nppiCrossCorrValid_NormLevel_8s32f_C1R_Ctx (C function), 1552
 nppiCrossCorrValid_NormLevel_8s32f_C3R_Ctx (C function), 1552
 nppiCrossCorrValid_NormLevel_8s32f_C4R_Ctx (C function), 1552
 nppiCrossCorrValid_NormLevel_8u32f_AC4R_Ctx (C function), 1551
 nppiCrossCorrValid_NormLevel_8u32f_C1R_Ctx (C function), 1551
 nppiCrossCorrValid_NormLevel_8u32f_C3R_Ctx (C function), 1551
 nppiCrossCorrValid_NormLevel_8u32f_C4R_Ctx (C function), 1551
 nppiCrossCorrValid_NormLevel_8u_AC4RSfs_Ctx (C function), 1549
 nppiCrossCorrValid_NormLevel_8u_C1RSfs_Ctx (C function), 1548
 nppiCrossCorrValid_NormLevel_8u_C3RSfs_Ctx (C function), 1549
 nppiCrossCorrValid_NormLevel_8u_C4RSfs_Ctx (C function), 1549
 nppiCrossCorrValid_NormLevel_16u32f_AC4R_Ctx (C function), 1553
 nppiCrossCorrValid_NormLevel_16u32f_C1R_Ctx (C function), 1553
 nppiCrossCorrValid_NormLevel_16u32f_C3R_Ctx (C function), 1553
 nppiCrossCorrValid_NormLevel_16u32f_C4R_Ctx (C function), 1553
 nppiCrossCorrValid_NormLevel_32f_AC4R_Ctx (C function), 1550
 nppiCrossCorrValid_NormLevel_32f_C1R_Ctx (C function), 1549
 nppiCrossCorrValid_NormLevel_32f_C3R_Ctx (C function), 1549
 nppiCrossCorrValid_NormLevel_32f_C4R_Ctx (C function), 1550
 nppiCrossCorrValid_NormLevel_64f_AC4R_Ctx (C function), 1551
 nppiCrossCorrValid_NormLevel_64f_C1R_Ctx (C function), 1550
 nppiCrossCorrValid_NormLevel_64f_C3R_Ctx (C function), 1550
 nppiCrossCorrValid_NormLevel_64f_C4R_Ctx (C function), 1550
 nppiCrossCorrValid_NormLevel_GetAdvancedSpadChBuf_8s32f_C3R_Ctx (C function), 1574
 nppiCrossCorrValid_NormLevelAdvanced_8s32f_AC4R_Ctx (C function), 1572
 nppiCrossCorrValid_NormLevelAdvanced_8s32f_C1R_Ctx (C function), 1572
 nppiCrossCorrValid_NormLevelAdvanced_8u32f_C1R_Ctx (C function), 1571
 nppiCrossCorrValid_NormLevelAdvanced_8u32f_C3R_Ctx (C function), 1571
 nppiCrossCorrValid_NormLevelAdvanced_8u32f_C4R_Ctx (C function), 1571
 nppiCrossCorrValid_NormLevelAdvanced_16u32f_C1R_Ctx (C function), 1573
 nppiCrossCorrValid_NormLevelAdvanced_16u32f_C3R_Ctx (C function), 1573
 nppiCrossCorrValid_NormLevelAdvanced_16u32f_C4R_Ctx (C function), 1573
 nppiCrossCorrValid_NormLevelAdvanced_32f_C1R_Ctx (C function), 1569
 nppiCrossCorrValid_NormLevelAdvanced_32f_C3R_Ctx (C function), 1569
 nppiCrossCorrValid_NormLevelAdvanced_32f_C4R_Ctx (C function), 1569
 nppiCrossCorrValid_NormLevelAdvanced_64f_C1R_Ctx (C function), 1570
 nppiCrossCorrValid_NormLevelAdvanced_64f_C3R_Ctx (C function), 1570
 nppiCrossCorrValid_NormLevelAdvanced_64f_C4R_Ctx (C function), 1570
 nppiDifferentialKernel (C enum), 23
 NppiDifferentialKernel.NPP_FILTER_SCHARR (C enumerator), 23
 NppiDifferentialKernel.NPP_FILTER_SOBEL (C enumerator), 23
 nppiDilate3x3_8u_AC4R_Ctx (C function), 1324
 nppiDilate3x3_8u_C1R_Ctx (C function), 1324
 nppiDilate3x3_8u_C3R_Ctx (C function), 1324
 nppiDilate3x3_8u_C4R_Ctx (C function), 1324
 nppiDilate3x3_16u_AC4R_Ctx (C function), 1325
 nppiDilate3x3_16u_C1R_Ctx (C function), 1324
 nppiDilate3x3_16u_C3R_Ctx (C function), 1324
 nppiDilate3x3_16u_C4R_Ctx (C function), 1324
 nppiDilate3x3_32f_AC4R_Ctx (C function), 1325
 nppiDilate3x3_32f_C1R_Ctx (C function), 1325
 nppiDilate3x3_32f_C3R_Ctx (C function), 1325
 nppiDilate3x3_32f_C4R_Ctx (C function), 1325
 nppiDilate3x3_64f_C1R_Ctx (C function), 1325

1325
 nppiDilate3x3Border_8u_AC4R_Ctx (C function), 1327
 nppiDilate3x3Border_8u_C1R_Ctx (C function), 1326
 nppiDilate3x3Border_8u_C3R_Ctx (C function), 1326
 nppiDilate3x3Border_8u_C4R_Ctx (C function), 1327
 nppiDilate3x3Border_16u_AC4R_Ctx (C function), 1327
 nppiDilate3x3Border_16u_C1R_Ctx (C function), 1327
 nppiDilate3x3Border_16u_C3R_Ctx (C function), 1327
 nppiDilate3x3Border_16u_C4R_Ctx (C function), 1327
 nppiDilate3x3Border_32f_AC4R_Ctx (C function), 1328
 nppiDilate3x3Border_32f_C1R_Ctx (C function), 1328
 nppiDilate3x3Border_32f_C3R_Ctx (C function), 1328
 nppiDilate3x3Border_32f_C4R_Ctx (C function), 1328
 nppiDilate_8u_AC4R_Ctx (C function), 1318
 nppiDilate_8u_C1R_Ctx (C function), 1318
 nppiDilate_8u_C3R_Ctx (C function), 1318
 nppiDilate_8u_C4R_Ctx (C function), 1318
 nppiDilate_16u_AC4R_Ctx (C function), 1319
 nppiDilate_16u_C1R_Ctx (C function), 1319
 nppiDilate_16u_C3R_Ctx (C function), 1319
 nppiDilate_16u_C4R_Ctx (C function), 1319
 nppiDilate_32f_AC4R_Ctx (C function), 1320
 nppiDilate_32f_C1R_Ctx (C function), 1319
 nppiDilate_32f_C3R_Ctx (C function), 1319
 nppiDilate_32f_C4R_Ctx (C function), 1319
 nppiDilateBorder_8u_AC4R_Ctx (C function), 1321
 nppiDilateBorder_8u_C1R_Ctx (C function), 1321
 nppiDilateBorder_8u_C3R_Ctx (C function), 1321
 nppiDilateBorder_8u_C4R_Ctx (C function), 1321
 nppiDilateBorder_16u_AC4R_Ctx (C function), 1322
 nppiDilateBorder_16u_C1R_Ctx (C function), 1321
 nppiDilateBorder_16u_C3R_Ctx (C function), 1322
 nppiDilateBorder_16u_C4R_Ctx (C function), 1322
 nppiDilateBorder_32f_AC4R_Ctx (C function), 1323
 nppiDilateBorder_32f_C1R_Ctx (C function), 1322
 nppiDilateBorder_32f_C3R_Ctx (C function), 1322
 nppiDilateBorder_32f_C4R_Ctx (C function), 1322
 nppiDistanceTransformAbsPBA_8s16u_C1R_Ctx (C function), 1129
 nppiDistanceTransformAbsPBA_8s32f_C1R_Ctx (C function), 1138
 nppiDistanceTransformAbsPBA_8s64f_C1R_Ctx (C function), 1147
 nppiDistanceTransformAbsPBA_8u16u_C1R_Ctx (C function), 1127
 nppiDistanceTransformAbsPBA_8u32f_C1R_Ctx (C function), 1136
 nppiDistanceTransformAbsPBA_8u64f_C1R_Ctx (C function), 1144
 nppiDistanceTransformAbsPBA_16s16u_C1R_Ctx (C function), 1133
 nppiDistanceTransformAbsPBA_16s32f_C1R_Ctx (C function), 1142
 nppiDistanceTransformAbsPBA_16s64f_C1R_Ctx (C function), 1151
 nppiDistanceTransformAbsPBA_16u16u_C1R_Ctx (C function), 1131
 nppiDistanceTransformAbsPBA_16u32f_C1R_Ctx (C function), 1140
 nppiDistanceTransformAbsPBA_16u64f_C1R_Ctx (C function), 1149
 nppiDistanceTransformAbsPBA_32f64f_C1R_Ctx (C function), 1154
 nppiDistanceTransformAbsPBA_64f_C1R_Ctx (C function), 1156
 nppiDistanceTransformPBA_8s16u_C1R_Ctx (C function), 1128
 nppiDistanceTransformPBA_8s32f_C1R_Ctx (C function), 1137
 nppiDistanceTransformPBA_8s64f_C1R_Ctx (C function), 1145
 nppiDistanceTransformPBA_8u16u_C1R_Ctx (C function), 1126
 nppiDistanceTransformPBA_8u32f_C1R_Ctx (C function), 1135
 nppiDistanceTransformPBA_8u64f_C1R_Ctx (C function), 1143
 nppiDistanceTransformPBA_16s16u_C1R_Ctx (C function), 1132
 nppiDistanceTransformPBA_16s32f_C1R_Ctx (C function), 1141
 nppiDistanceTransformPBA_16s64f_C1R_Ctx (C function), 1150
 nppiDistanceTransformPBA_16u16u_C1R_Ctx

(C function), 1130
 nppiDistanceTransformPBA_16u32f_C1R_Ctx
 (C function), 1139
 nppiDistanceTransformPBA_16u64f_C1R_Ctx
 (C function), 1148
 nppiDistanceTransformPBA_32f64f_C1R_Ctx
 (C function), 1153
 nppiDistanceTransformPBA_64f_C1R_Ctx (C
 function), 1155
 nppiDistanceTransformPBAGetAntialiasingBufferSize
 (C function), 1125
 nppiDistanceTransformPBAGetBufferSize
 (C function), 1125
 nppiDiv_8u_AC4IRSfs_Ctx (C function), 329
 nppiDiv_8u_AC4RSfs_Ctx (C function), 329
 nppiDiv_8u_C1IRSfs_Ctx (C function), 328
 nppiDiv_8u_C1RSfs_Ctx (C function), 327
 nppiDiv_8u_C3IRSfs_Ctx (C function), 329
 nppiDiv_8u_C3RSfs_Ctx (C function), 328
 nppiDiv_8u_C4IRSfs_Ctx (C function), 330
 nppiDiv_8u_C4RSfs_Ctx (C function), 330
 nppiDiv_16f_C1IR_Ctx (C function), 346
 nppiDiv_16f_C1R_Ctx (C function), 346
 nppiDiv_16f_C3IR_Ctx (C function), 347
 nppiDiv_16f_C3R_Ctx (C function), 346
 nppiDiv_16f_C4IR_Ctx (C function), 347
 nppiDiv_16f_C4R_Ctx (C function), 347
 nppiDiv_16s_AC4IRSfs_Ctx (C function), 337
 nppiDiv_16s_AC4RSfs_Ctx (C function), 336
 nppiDiv_16s_C1IRSfs_Ctx (C function), 335
 nppiDiv_16s_C1RSfs_Ctx (C function), 335
 nppiDiv_16s_C3IRSfs_Ctx (C function), 336
 nppiDiv_16s_C3RSfs_Ctx (C function), 336
 nppiDiv_16s_C4IRSfs_Ctx (C function), 338
 nppiDiv_16s_C4RSfs_Ctx (C function), 337
 nppiDiv_16sc_AC4IRSfs_Ctx (C function), 340
 nppiDiv_16sc_AC4RSfs_Ctx (C function), 340
 nppiDiv_16sc_C1IRSfs_Ctx (C function), 339
 nppiDiv_16sc_C1RSfs_Ctx (C function), 338
 nppiDiv_16sc_C3IRSfs_Ctx (C function), 339
 nppiDiv_16sc_C3RSfs_Ctx (C function), 339
 nppiDiv_16u_AC4IRSfs_Ctx (C function), 333
 nppiDiv_16u_AC4RSfs_Ctx (C function), 333
 nppiDiv_16u_C1IRSfs_Ctx (C function), 332
 nppiDiv_16u_C1RSfs (C function), 331
 nppiDiv_16u_C1RSfs_Ctx (C function), 331
 nppiDiv_16u_C3IRSfs_Ctx (C function), 332
 nppiDiv_16u_C3RSfs_Ctx (C function), 332
 nppiDiv_16u_C4IRSfs_Ctx (C function), 334
 nppiDiv_16u_C4RSfs_Ctx (C function), 334
 nppiDiv_32f_AC4IR_Ctx (C function), 350
 nppiDiv_32f_AC4R_Ctx (C function), 349
 nppiDiv_32f_C1IR_Ctx (C function), 348
 nppiDiv_32f_C1R_Ctx (C function), 348
 nppiDiv_32f_C3IR_Ctx (C function), 349
 nppiDiv_32f_C3R_Ctx (C function), 349
 nppiDiv_32f_C4IR_Ctx (C function), 351
 nppiDiv_32f_C4R_Ctx (C function), 350
 nppiDiv_32fc_AC4IR_Ctx (C function), 353
 nppiDiv_32fc_AC4R_Ctx (C function), 352
 nppiDiv_32fc_C1IR_Ctx (C function), 351
 nppiDiv_32fc_C1R_Ctx (C function), 351
 nppiDiv_32fc_C3IR_Ctx (C function), 352
 nppiDiv_32fc_C3R_Ctx (C function), 352
 nppiDiv_32fc_C4IR_Ctx (C function), 354
 nppiDiv_32fc_C4R_Ctx (C function), 353
 nppiDiv_32s_C1IRSfs_Ctx (C function), 342
 nppiDiv_32s_C1R_Ctx (C function), 341
 nppiDiv_32s_C1RSfs_Ctx (C function), 341
 nppiDiv_32s_C3IRSfs_Ctx (C function), 343
 nppiDiv_32s_C3RSfs_Ctx (C function), 342
 nppiDiv_32sc_AC4IRSfs_Ctx (C function), 345
 nppiDiv_32sc_AC4RSfs_Ctx (C function), 345
 nppiDiv_32sc_C1IRSfs_Ctx (C function), 343
 nppiDiv_32sc_C1RSfs_Ctx (C function), 343
 nppiDiv_32sc_C3IRSfs_Ctx (C function), 344
 nppiDiv_32sc_C3RSfs_Ctx (C function), 344
 nppiDiv_Round_8u_AC4IRSfs_Ctx (C func-
 tion), 357
 nppiDiv_Round_8u_AC4RSfs_Ctx (C function),
 356
 nppiDiv_Round_8u_C1IRSfs_Ctx (C function),
 355
 nppiDiv_Round_8u_C1RSfs_Ctx (C function),
 354
 nppiDiv_Round_8u_C3IRSfs_Ctx (C function),
 356
 nppiDiv_Round_8u_C3RSfs_Ctx (C function),
 355
 nppiDiv_Round_8u_C4IRSfs_Ctx (C function),
 358
 nppiDiv_Round_8u_C4RSfs_Ctx (C function),
 357
 nppiDiv_Round_16s_AC4IRSfs_Ctx (C func-
 tion), 365
 nppiDiv_Round_16s_AC4RSfs_Ctx (C func-
 tion), 364
 nppiDiv_Round_16s_C1IRSfs_Ctx (C func-
 tion), 363
 nppiDiv_Round_16s_C1RSfs_Ctx (C function),
 362
 nppiDiv_Round_16s_C3IRSfs_Ctx (C func-
 tion), 364
 nppiDiv_Round_16s_C3RSfs_Ctx (C function),
 363
 nppiDiv_Round_16s_C4IRSfs_Ctx (C func-
 tion), 366
 nppiDiv_Round_16s_C4RSfs_Ctx (C function),

365
 nppiDiv_Round_16u_AC4IRSfs_Ctx (C function), 361
 nppiDiv_Round_16u_AC4RSfs_Ctx (C function), 360
 nppiDiv_Round_16u_C1IRSfs_Ctx (C function), 359
 nppiDiv_Round_16u_C1RSfs_Ctx (C function), 358
 nppiDiv_Round_16u_C3IRSfs_Ctx (C function), 360
 nppiDiv_Round_16u_C3RSfs_Ctx (C function), 359
 nppiDiv_Round_16u_C4IRSfs_Ctx (C function), 362
 nppiDiv_Round_16u_C4RSfs_Ctx (C function), 361
 nppiDivC_8u_AC4IRSfs_Ctx (C function), 191
 nppiDivC_8u_AC4RSfs_Ctx (C function), 190
 nppiDivC_8u_C1IRSfs_Ctx (C function), 188
 nppiDivC_8u_C1RSfs_Ctx (C function), 187
 nppiDivC_8u_C3IRSfs_Ctx (C function), 190
 nppiDivC_8u_C3RSfs_Ctx (C function), 189
 nppiDivC_8u_C4IRSfs_Ctx (C function), 193
 nppiDivC_8u_C4RSfs_Ctx (C function), 192
 nppiDivC_16f_C1IR_Ctx (C function), 216
 nppiDivC_16f_C1R_Ctx (C function), 215
 nppiDivC_16f_C3IR_Ctx (C function), 217
 nppiDivC_16f_C3R_Ctx (C function), 217
 nppiDivC_16f_C4IR_Ctx (C function), 219
 nppiDivC_16f_C4R_Ctx (C function), 218
 nppiDivC_16s_AC4IRSfs_Ctx (C function), 204
 nppiDivC_16s_AC4RSfs_Ctx (C function), 203
 nppiDivC_16s_C1IRSfs_Ctx (C function), 201
 nppiDivC_16s_C1RSfs_Ctx (C function), 200
 nppiDivC_16s_C3IRSfs_Ctx (C function), 203
 nppiDivC_16s_C3RSfs_Ctx (C function), 202
 nppiDivC_16s_C4IRSfs_Ctx (C function), 206
 nppiDivC_16s_C4RSfs_Ctx (C function), 205
 nppiDivC_16sc_AC4IRSfs_Ctx (C function), 209
 nppiDivC_16sc_AC4RSfs_Ctx (C function), 208
 nppiDivC_16sc_C1IRSfs_Ctx (C function), 207
 nppiDivC_16sc_C1RSfs_Ctx (C function), 207
 nppiDivC_16sc_C3IRSfs_Ctx (C function), 208
 nppiDivC_16sc_C3RSfs_Ctx (C function), 208
 nppiDivC_16u_AC4IRSfs_Ctx (C function), 198
 nppiDivC_16u_AC4RSfs_Ctx (C function), 197
 nppiDivC_16u_C1IRSfs_Ctx (C function), 194
 nppiDivC_16u_C1RSfs_Ctx (C function), 194
 nppiDivC_16u_C3IRSfs_Ctx (C function), 196
 nppiDivC_16u_C3RSfs_Ctx (C function), 195
 nppiDivC_16u_C4IRSfs_Ctx (C function), 199
 nppiDivC_16u_C4RSfs_Ctx (C function), 199
 nppiDivC_32f_AC4IR_Ctx (C function), 223
 nppiDivC_32f_AC4R_Ctx (C function), 222
 nppiDivC_32f_C1IR_Ctx (C function), 220
 nppiDivC_32f_C1R_Ctx (C function), 220
 nppiDivC_32f_C3IR_Ctx (C function), 222
 nppiDivC_32f_C3R_Ctx (C function), 221
 nppiDivC_32f_C4IR_Ctx (C function), 225
 nppiDivC_32f_C4R_Ctx (C function), 224
 nppiDivC_32fc_AC4IR_Ctx (C function), 227
 nppiDivC_32fc_AC4R_Ctx (C function), 227
 nppiDivC_32fc_C1IR_Ctx (C function), 226
 nppiDivC_32fc_C1R_Ctx (C function), 225
 nppiDivC_32fc_C3IR_Ctx (C function), 226
 nppiDivC_32fc_C3R_Ctx (C function), 226
 nppiDivC_32fc_C4IR_Ctx (C function), 228
 nppiDivC_32fc_C4R_Ctx (C function), 228
 nppiDivC_32s_C1IRSfs_Ctx (C function), 210
 nppiDivC_32s_C1RSfs (C function), 210
 nppiDivC_32s_C1RSfs_Ctx (C function), 209
 nppiDivC_32s_C3IRSfs_Ctx (C function), 212
 nppiDivC_32s_C3RSfs_Ctx (C function), 211
 nppiDivC_32sc_AC4IRSfs_Ctx (C function), 215
 nppiDivC_32sc_AC4RSfs_Ctx (C function), 214
 nppiDivC_32sc_C1IRSfs_Ctx (C function), 213
 nppiDivC_32sc_C1RSfs_Ctx (C function), 213
 nppiDivC_32sc_C3IRSfs_Ctx (C function), 214
 nppiDivC_32sc_C3RSfs_Ctx (C function), 214
 nppiDivDeviceC_8u_AC4IRSfs_Ctx (C function), 192
 nppiDivDeviceC_8u_AC4RSfs_Ctx (C function), 191
 nppiDivDeviceC_8u_C1IRSfs_Ctx (C function), 188
 nppiDivDeviceC_8u_C1RSfs_Ctx (C function), 187
 nppiDivDeviceC_8u_C3IRSfs_Ctx (C function), 190
 nppiDivDeviceC_8u_C3RSfs_Ctx (C function), 189
 nppiDivDeviceC_8u_C4IRSfs_Ctx (C function), 193
 nppiDivDeviceC_8u_C4RSfs_Ctx (C function), 192
 nppiDivDeviceC_16f_C1IR_Ctx (C function), 216
 nppiDivDeviceC_16f_C1R_Ctx (C function), 216
 nppiDivDeviceC_16f_C3IR_Ctx (C function), 218
 nppiDivDeviceC_16f_C3R_Ctx (C function), 217
 nppiDivDeviceC_16f_C4IR_Ctx (C function), 219

nppiDivDeviceC_16f_C4R_Ctx (C function), 219	nppiDivDeviceC_32s_C3IRSfs_Ctx (C func- tion), 212
nppiDivDeviceC_16s_AC4IRSfs_Ctx (C func- tion), 205	nppiDivDeviceC_32s_C3RSfs_Ctx (C func- tion), 212
nppiDivDeviceC_16s_AC4RSfs_Ctx (C func- tion), 204	nppiDotProd_8s64f_AC4R_Ctx (C function), 1469
nppiDivDeviceC_16s_C1IRSfs_Ctx (C func- tion), 201	nppiDotProd_8s64f_C1R_Ctx (C function), 1466
nppiDivDeviceC_16s_C1RSfs_Ctx (C func- tion), 201	nppiDotProd_8s64f_C3R_Ctx (C function), 1467
nppiDivDeviceC_16s_C3IRSfs_Ctx (C func- tion), 203	nppiDotProd_8s64f_C4R_Ctx (C function), 1468
nppiDivDeviceC_16s_C3RSfs_Ctx (C func- tion), 202	nppiDotProd_8u64f_AC4R_Ctx (C function), 1469
nppiDivDeviceC_16s_C4IRSfs_Ctx (C func- tion), 206	nppiDotProd_8u64f_C1R_Ctx (C function), 1466
nppiDivDeviceC_16s_C4RSfs_Ctx (C func- tion), 206	nppiDotProd_8u64f_C3R_Ctx (C function), 1467
nppiDivDeviceC_16u_AC4IRSfs_Ctx (C func- tion), 198	nppiDotProd_8u64f_C4R_Ctx (C function), 1468
nppiDivDeviceC_16u_AC4RSfs_Ctx (C func- tion), 197	nppiDotProd_16s64f_AC4R_Ctx (C function), 1470
nppiDivDeviceC_16u_C1IRSfs_Ctx (C func- tion), 195	nppiDotProd_16s64f_C1R_Ctx (C function), 1467
nppiDivDeviceC_16u_C1RSfs_Ctx (C func- tion), 194	nppiDotProd_16s64f_C3R_Ctx (C function), 1468
nppiDivDeviceC_16u_C3IRSfs_Ctx (C func- tion), 196	nppiDotProd_16s64f_C4R_Ctx (C function), 1469
nppiDivDeviceC_16u_C3RSfs_Ctx (C func- tion), 196	nppiDotProd_16u64f_AC4R_Ctx (C function), 1469
nppiDivDeviceC_16u_C4IRSfs_Ctx (C func- tion), 200	nppiDotProd_16u64f_C1R_Ctx (C function), 1467
nppiDivDeviceC_16u_C4RSfs_Ctx (C func- tion), 199	nppiDotProd_16u64f_C3R_Ctx (C function), 1468
nppiDivDeviceC_32f_AC4IR_Ctx (C function), 224	nppiDotProd_16u64f_C4R_Ctx (C function), 1468
nppiDivDeviceC_32f_AC4R_Ctx (C function), 223	nppiDotProd_32f64f_AC4R_Ctx (C function), 1470
nppiDivDeviceC_32f_C1IR_Ctx (C function), 221	nppiDotProd_32f64f_C1R_Ctx (C function), 1467
nppiDivDeviceC_32f_C1R_Ctx (C function), 220	nppiDotProd_32f64f_C3R_Ctx (C function), 1468
nppiDivDeviceC_32f_C3IR_Ctx (C function), 222	nppiDotProd_32f64f_C4R_Ctx (C function), 1469
nppiDivDeviceC_32f_C3R_Ctx (C function), 221	nppiDotProd_32s64f_AC4R_Ctx (C function), 1470
nppiDivDeviceC_32f_C4IR_Ctx (C function), 225	nppiDotProd_32s64f_C1R_Ctx (C function), 1467
nppiDivDeviceC_32f_C4R_Ctx (C function), 224	nppiDotProd_32s64f_C3R_Ctx (C function), 1468
nppiDivDeviceC_32s_C1IRSfs_Ctx (C func- tion), 211	nppiDotProd_32s64f_C4R_Ctx (C function), 1469
nppiDivDeviceC_32s_C1RSfs_Ctx (C func- tion), 210	nppiDotProd_32u64f_AC4R_Ctx (C function), 1470

nppiDotProd_32u64f_C1R_Ctx (C function), 1467
 nppiDotProd_32u64f_C3R_Ctx (C function), 1468
 nppiDotProd_32u64f_C4R_Ctx (C function), 1469
 nppiDotProdGetBufferHostSize_8s64f_AC4R_Ctx (C function), 1473
 nppiDotProdGetBufferHostSize_8s64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferHostSize_8s64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferHostSize_8s64f_C4R_Ctx (C function), 1472
 nppiDotProdGetBufferHostSize_8u64f_AC4R_Ctx (C function), 1473
 nppiDotProdGetBufferHostSize_8u64f_C1R_Ctx (C function), 1470
 nppiDotProdGetBufferHostSize_8u64f_C3R_Ctx (C function), 1471
 nppiDotProdGetBufferHostSize_8u64f_C4R_Ctx (C function), 1472
 nppiDotProdGetBufferHostSize_16s64f_AC4R_Ctx (C function), 1474
 nppiDotProdGetBufferHostSize_16s64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferHostSize_16s64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferHostSize_16s64f_C4R_Ctx (C function), 1473
 nppiDotProdGetBufferHostSize_16u64f_AC4R_Ctx (C function), 1474
 nppiDotProdGetBufferHostSize_16u64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferHostSize_16u64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferHostSize_16u64f_C4R_Ctx (C function), 1473
 nppiDotProdGetBufferHostSize_32f64f_AC4R_Ctx (C function), 1474
 nppiDotProdGetBufferHostSize_32f64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferHostSize_32f64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferHostSize_32f64f_C4R_Ctx (C function), 1473
 nppiDotProdGetBufferHostSize_32s64f_AC4R_Ctx (C function), 1474
 nppiDotProdGetBufferHostSize_32s64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferHostSize_32s64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferHostSize_32s64f_C4R_Ctx (C function), 1473
 nppiDotProdGetBufferSize_32u64f_AC4R_Ctx (C function), 1474
 nppiDotProdGetBufferSize_32u64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferSize_32u64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferSize_32u64f_C4R_Ctx (C function), 1472
 nppiDotProdGetBufferSize_8s64f_AC4R_Ctx (C function), 1473
 nppiDotProdGetBufferSize_8s64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferSize_8s64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferSize_8s64f_C4R_Ctx (C function), 1472
 nppiDotProdGetBufferSize_8u64f_AC4R_Ctx (C function), 1473
 nppiDotProdGetBufferSize_8u64f_C1R_Ctx (C function), 1470
 nppiDotProdGetBufferSize_8u64f_C3R_Ctx (C function), 1471
 nppiDotProdGetBufferSize_8u64f_C4R_Ctx (C function), 1472
 nppiDotProdGetBufferSize_16s64f_AC4R_Ctx (C function), 1474
 nppiDotProdGetBufferSize_16s64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferSize_16s64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferSize_16s64f_C4R_Ctx (C function), 1473
 nppiDotProdGetBufferSize_16u64f_AC4R_Ctx (C function), 1474
 nppiDotProdGetBufferSize_16u64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferSize_16u64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferSize_16u64f_C4R_Ctx (C function), 1473
 nppiDotProdGetBufferSize_32f64f_AC4R_Ctx (C function), 1474
 nppiDotProdGetBufferSize_32f64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferSize_32f64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferSize_32f64f_C4R_Ctx (C function), 1473
 nppiDotProdGetBufferSize_32s64f_AC4R_Ctx (C function), 1474
 nppiDotProdGetBufferSize_32s64f_C1R_Ctx (C function), 1471
 nppiDotProdGetBufferSize_32s64f_C3R_Ctx (C function), 1472
 nppiDotProdGetBufferSize_32s64f_C4R_Ctx (C function), 1473
 nppiDup_8u_C1AC4R_Ctx (C function), 876
 nppiDup_8u_C1C3R_Ctx (C function), 876
 nppiDup_8u_C1C4R_Ctx (C function), 876
 nppiDup_16s_C1AC4R_Ctx (C function), 877
 nppiDup_16s_C1C3R_Ctx (C function), 876
 nppiDup_16s_C1C4R_Ctx (C function), 876
 nppiDup_16u_C1AC4R_Ctx (C function), 876
 nppiDup_16u_C1C3R_Ctx (C function), 876
 nppiDup_16u_C1C4R_Ctx (C function), 876
 nppiDup_32f_C1AC4R_Ctx (C function), 877
 nppiDup_32f_C1C3R_Ctx (C function), 877
 nppiDup_32f_C1C4R_Ctx (C function), 877
 nppiDup_32s_C1AC4R_Ctx (C function), 877
 nppiDup_32s_C1C3R_Ctx (C function), 877
 nppiDup_32s_C1C4R_Ctx (C function), 877
 nppiErode3x3_8u_AC4R_Ctx (C function), 1335
 nppiErode3x3_8u_C1R_Ctx (C function), 1335
 nppiErode3x3_8u_C3R_Ctx (C function), 1335
 nppiErode3x3_8u_C4R_Ctx (C function), 1335
 nppiErode3x3_16u_AC4R_Ctx (C function), 1335
 nppiErode3x3_16u_C1R_Ctx (C function), 1335
 nppiErode3x3_16u_C3R_Ctx (C function), 1335
 nppiErode3x3_16u_C4R_Ctx (C function), 1335
 nppiErode3x3_32f_AC4R_Ctx (C function), 1336
 nppiErode3x3_32f_C1R_Ctx (C function), 1336
 nppiErode3x3_32f_C3R_Ctx (C function), 1336
 nppiErode3x3_32f_C4R_Ctx (C function), 1336
 nppiErode3x3_64f_C1R_Ctx (C function), 1336
 nppiErode3x3Border_8u_AC4R_Ctx (C function), 1338
 nppiErode3x3Border_8u_C1R_Ctx (C function), 1337
 nppiErode3x3Border_8u_C3R_Ctx (C function), 1337
 nppiErode3x3Border_8u_C4R_Ctx (C function), 1337
 nppiErode3x3Border_16u_AC4R_Ctx (C function), 1338
 nppiErode3x3Border_16u_C1R_Ctx (C function), 1338
 nppiErode3x3Border_16u_C3R_Ctx (C function), 1338
 nppiErode3x3Border_16u_C4R_Ctx (C function), 1338

nppiErode3x3Border_32f_AC4R_Ctx (C function), 1339
 nppiErode3x3Border_32f_C1R_Ctx (C function), 1338
 nppiErode3x3Border_32f_C3R_Ctx (C function), 1339
 nppiErode3x3Border_32f_C4R_Ctx (C function), 1339
 nppiErode_8u_AC4R_Ctx (C function), 1329
 nppiErode_8u_C1R_Ctx (C function), 1329
 nppiErode_8u_C3R_Ctx (C function), 1329
 nppiErode_8u_C4R_Ctx (C function), 1329
 nppiErode_16u_AC4R_Ctx (C function), 1330
 nppiErode_16u_C1R_Ctx (C function), 1330
 nppiErode_16u_C3R_Ctx (C function), 1330
 nppiErode_16u_C4R_Ctx (C function), 1330
 nppiErode_32f_AC4R_Ctx (C function), 1331
 nppiErode_32f_C1R_Ctx (C function), 1330
 nppiErode_32f_C3R_Ctx (C function), 1330
 nppiErode_32f_C4R_Ctx (C function), 1330
 nppiErodeBorder_8u_AC4R_Ctx (C function), 1332
 nppiErodeBorder_8u_C1R_Ctx (C function), 1332
 nppiErodeBorder_8u_C3R_Ctx (C function), 1332
 nppiErodeBorder_8u_C4R_Ctx (C function), 1332
 nppiErodeBorder_16u_AC4R_Ctx (C function), 1333
 nppiErodeBorder_16u_C1R_Ctx (C function), 1332
 nppiErodeBorder_16u_C3R_Ctx (C function), 1333
 nppiErodeBorder_16u_C4R_Ctx (C function), 1333
 nppiErodeBorder_32f_AC4R_Ctx (C function), 1333
 nppiErodeBorder_32f_C1R_Ctx (C function), 1333
 nppiErodeBorder_32f_C3R_Ctx (C function), 1333
 nppiErodeBorder_32f_C4R_Ctx (C function), 1333
 nppiEvenLevelsHost_32s (C function), 1489
 nppiExp_8u_C1RSfs_Ctx (C function), 407
 nppiExp_8u_C1RSfs_Ctx (C function), 407
 nppiExp_8u_C3RSfs_Ctx (C function), 408
 nppiExp_8u_C3RSfs_Ctx (C function), 407
 nppiExp_16s_C1RSfs_Ctx (C function), 410
 nppiExp_16s_C1RSfs_Ctx (C function), 410
 nppiExp_16s_C3RSfs_Ctx (C function), 411
 nppiExp_16s_C3RSfs_Ctx (C function), 410
 nppiExp_16u_C1RSfs_Ctx (C function), 408
 nppiExp_16u_C1RSfs_Ctx (C function), 408
 nppiExp_16u_C3RSfs_Ctx (C function), 409
 nppiExp_16u_C3RSfs_Ctx (C function), 409
 nppiExp_32f_C1IR_Ctx (C function), 411
 nppiExp_32f_C1R_Ctx (C function), 411
 nppiExp_32f_C3IR_Ctx (C function), 412
 nppiExp_32f_C3R_Ctx (C function), 412
 nppiFilter32f_8s16s_AC4R_Ctx (C function), 948
 nppiFilter32f_8s16s_C1R_Ctx (C function), 948
 nppiFilter32f_8s16s_C3R_Ctx (C function), 948
 nppiFilter32f_8s16s_C4R_Ctx (C function), 948
 nppiFilter32f_8s_AC4R_Ctx (C function), 945
 nppiFilter32f_8s_C1R_Ctx (C function), 944
 nppiFilter32f_8s_C2R_Ctx (C function), 945
 nppiFilter32f_8s_C3R_Ctx (C function), 945
 nppiFilter32f_8s_C4R_Ctx (C function), 945
 nppiFilter32f_8u16s_AC4R_Ctx (C function), 948
 nppiFilter32f_8u16s_C1R_Ctx (C function), 947
 nppiFilter32f_8u16s_C3R_Ctx (C function), 947
 nppiFilter32f_8u16s_C4R_Ctx (C function), 947
 nppiFilter32f_8u_AC4R_Ctx (C function), 944
 nppiFilter32f_8u_C1R_Ctx (C function), 944
 nppiFilter32f_8u_C2R_Ctx (C function), 944
 nppiFilter32f_8u_C3R_Ctx (C function), 944
 nppiFilter32f_8u_C4R_Ctx (C function), 944
 nppiFilter32f_16f_C1R_Ctx (C function), 947
 nppiFilter32f_16f_C3R_Ctx (C function), 947
 nppiFilter32f_16f_C4R_Ctx (C function), 947
 nppiFilter32f_16s_AC4R_Ctx (C function), 946
 nppiFilter32f_16s_C1R_Ctx (C function), 946
 nppiFilter32f_16s_C3R_Ctx (C function), 946
 nppiFilter32f_16s_C4R_Ctx (C function), 946
 nppiFilter32f_16u_AC4R_Ctx (C function), 946
 nppiFilter32f_16u_C1R_Ctx (C function), 945
 nppiFilter32f_16u_C3R_Ctx (C function), 945
 nppiFilter32f_16u_C4R_Ctx (C function), 945
 nppiFilter32f_32s_AC4R_Ctx (C function), 947
 nppiFilter32f_32s_C1R_Ctx (C function), 946
 nppiFilter32f_32s_C3R_Ctx (C function), 946
 nppiFilter32f_32s_C4R_Ctx (C function), 947
 nppiFilter_8u_AC4R_Ctx (C function), 941
 nppiFilter_8u_C1R_Ctx (C function), 941
 nppiFilter_8u_C3R_Ctx (C function), 941

nppiFilter_8u_C4R_Ctx (C function), 941
 nppiFilter_16s_AC4R_Ctx (C function), 942
 nppiFilter_16s_C1R_Ctx (C function), 942
 nppiFilter_16s_C3R_Ctx (C function), 942
 nppiFilter_16s_C4R_Ctx (C function), 942
 nppiFilter_16u_AC4R_Ctx (C function), 941
 nppiFilter_16u_C1R_Ctx (C function), 941
 nppiFilter_16u_C3R_Ctx (C function), 941
 nppiFilter_16u_C4R_Ctx (C function), 941
 nppiFilter_32f_AC4R_Ctx (C function), 943
 nppiFilter_32f_C1R_Ctx (C function), 942
 nppiFilter_32f_C2R_Ctx (C function), 942
 nppiFilter_32f_C3R_Ctx (C function), 942
 nppiFilter_32f_C4R_Ctx (C function), 943
 nppiFilter_64f_C1R_Ctx (C function), 943
 nppiFilterBilateralGaussBorder_8u_C1R_Ctx (C function), 1073
 nppiFilterBilateralGaussBorder_8u_C3R_Ctx (C function), 1073
 nppiFilterBilateralGaussBorder_16u_C1R_Ctx (C function), 1073
 nppiFilterBilateralGaussBorder_16u_C3R_Ctx (C function), 1073
 nppiFilterBilateralGaussBorder_32f_C1R_Ctx (C function), 1074
 nppiFilterBilateralGaussBorder_32f_C3R_Ctx (C function), 1074
 nppiFilterBorder32f_8s16s_AC4R_Ctx (C function), 959
 nppiFilterBorder32f_8s16s_C1R_Ctx (C function), 958
 nppiFilterBorder32f_8s16s_C3R_Ctx (C function), 959
 nppiFilterBorder32f_8s16s_C4R_Ctx (C function), 959
 nppiFilterBorder32f_8s_AC4R_Ctx (C function), 955
 nppiFilterBorder32f_8s_C1R_Ctx (C function), 954
 nppiFilterBorder32f_8s_C2R_Ctx (C function), 954
 nppiFilterBorder32f_8s_C3R_Ctx (C function), 955
 nppiFilterBorder32f_8s_C4R_Ctx (C function), 955
 nppiFilterBorder32f_8u16s_AC4R_Ctx (C function), 958
 nppiFilterBorder32f_8u16s_C1R_Ctx (C function), 958
 nppiFilterBorder32f_8u16s_C3R_Ctx (C function), 958
 nppiFilterBorder32f_8u16s_C4R_Ctx (C function), 958
 nppiFilterBorder32f_8u_AC4R_Ctx (C function), 954
 nppiFilterBorder32f_8u_C1R_Ctx (C function), 953
 nppiFilterBorder32f_8u_C2R_Ctx (C function), 953
 nppiFilterBorder32f_8u_C3R_Ctx (C function), 954
 nppiFilterBorder32f_8u_C4R_Ctx (C function), 954
 nppiFilterBorder32f_16f_C1R_Ctx (C function), 957
 nppiFilterBorder32f_16f_C3R_Ctx (C function), 957
 nppiFilterBorder32f_16f_C4R_Ctx (C function), 958
 nppiFilterBorder32f_16s_AC4R_Ctx (C function), 956
 nppiFilterBorder32f_16s_C1R_Ctx (C function), 956
 nppiFilterBorder32f_16s_C3R_Ctx (C function), 956
 nppiFilterBorder32f_16s_C4R_Ctx (C function), 956
 nppiFilterBorder32f_16u_AC4R_Ctx (C function), 956
 nppiFilterBorder32f_16u_C1R_Ctx (C function), 955
 nppiFilterBorder32f_16u_C3R_Ctx (C function), 955
 nppiFilterBorder32f_16u_C4R_Ctx (C function), 955
 nppiFilterBorder32f_32s_AC4R_Ctx (C function), 957
 nppiFilterBorder32f_32s_C1R_Ctx (C function), 956
 nppiFilterBorder32f_32s_C3R_Ctx (C function), 957
 nppiFilterBorder32f_32s_C4R_Ctx (C function), 957
 nppiFilterBorder_8u_AC4R_Ctx (C function), 950
 nppiFilterBorder_8u_C1R_Ctx (C function), 949
 nppiFilterBorder_8u_C3R_Ctx (C function), 949
 nppiFilterBorder_8u_C4R_Ctx (C function), 950
 nppiFilterBorder_16s_AC4R_Ctx (C function), 951
 nppiFilterBorder_16s_C1R_Ctx (C function), 951
 nppiFilterBorder_16s_C3R_Ctx (C function), 951
 nppiFilterBorder_16s_C4R_Ctx (C function), 951

[951](#)
 nppiFilterBorder_16u_AC4R_Ctx (C function), [951](#)
 nppiFilterBorder_16u_C1R_Ctx (C function), [950](#)
 nppiFilterBorder_16u_C3R_Ctx (C function), [950](#)
 nppiFilterBorder_16u_C4R_Ctx (C function), [950](#)
 nppiFilterBorder_32f_AC4R_Ctx (C function), [952](#)
 nppiFilterBorder_32f_C1R_Ctx (C function), [951](#)
 nppiFilterBorder_32f_C2R_Ctx (C function), [952](#)
 nppiFilterBorder_32f_C3R_Ctx (C function), [952](#)
 nppiFilterBorder_32f_C4R_Ctx (C function), [952](#)
 nppiFilterBox_8u_AC4R_Ctx (C function), [961](#)
 nppiFilterBox_8u_C1R_Ctx (C function), [960](#)
 nppiFilterBox_8u_C3R_Ctx (C function), [960](#)
 nppiFilterBox_8u_C4R_Ctx (C function), [960](#)
 nppiFilterBox_16s_AC4R_Ctx (C function), [962](#)
 nppiFilterBox_16s_C1R_Ctx (C function), [961](#)
 nppiFilterBox_16s_C3R_Ctx (C function), [961](#)
 nppiFilterBox_16s_C4R_Ctx (C function), [962](#)
 nppiFilterBox_16u_AC4R_Ctx (C function), [961](#)
 nppiFilterBox_16u_C1R_Ctx (C function), [961](#)
 nppiFilterBox_16u_C3R_Ctx (C function), [961](#)
 nppiFilterBox_16u_C4R_Ctx (C function), [961](#)
 nppiFilterBox_32f_AC4R_Ctx (C function), [962](#)
 nppiFilterBox_32f_C1R_Ctx (C function), [962](#)
 nppiFilterBox_32f_C3R_Ctx (C function), [962](#)
 nppiFilterBox_32f_C4R_Ctx (C function), [962](#)
 nppiFilterBox_64f_C1R_Ctx (C function), [962](#)
 nppiFilterBoxBorder_8u_AC4R_Ctx (C function), [964](#)
 nppiFilterBoxBorder_8u_C1R_Ctx (C function), [964](#)
 nppiFilterBoxBorder_8u_C3R_Ctx (C function), [964](#)
 nppiFilterBoxBorder_8u_C4R_Ctx (C function), [964](#)
 nppiFilterBoxBorder_16s_AC4R_Ctx (C function), [965](#)
 nppiFilterBoxBorder_16s_C1R_Ctx (C function), [965](#)
 nppiFilterBoxBorder_16s_C3R_Ctx (C function), [965](#)
 nppiFilterBoxBorder_16s_C4R_Ctx (C function), [965](#)
 nppiFilterBoxBorder_16u_AC4R_Ctx (C function), [965](#)
 nppiFilterBoxBorder_16u_C1R_Ctx (C function), [965](#)
 nppiFilterBoxBorder_16u_C3R_Ctx (C function), [964](#)
 nppiFilterBoxBorder_16u_C4R_Ctx (C function), [964](#)
 nppiFilterBoxBorder_32f_AC4R_Ctx (C function), [966](#)
 nppiFilterBoxBorder_32f_C1R_Ctx (C function), [966](#)
 nppiFilterBoxBorder_32f_C3R_Ctx (C function), [966](#)
 nppiFilterBoxBorder_32f_C4R_Ctx (C function), [966](#)
 nppiFilterBoxBorderAdvanced_8u_C1R_Ctx (C function), [968](#)
 nppiFilterBoxBorderAdvanced_8u_C3R_Ctx (C function), [968](#)
 nppiFilterBoxBorderAdvanced_8u_C4R_Ctx (C function), [968](#)
 nppiFilterBoxBorderAdvanced_16s_C1R_Ctx (C function), [969](#)
 nppiFilterBoxBorderAdvanced_16s_C3R_Ctx (C function), [969](#)
 nppiFilterBoxBorderAdvanced_16s_C4R_Ctx (C function), [970](#)
 nppiFilterBoxBorderAdvanced_16u_C1R_Ctx (C function), [969](#)
 nppiFilterBoxBorderAdvanced_16u_C3R_Ctx (C function), [969](#)
 nppiFilterBoxBorderAdvanced_16u_C4R_Ctx (C function), [969](#)
 nppiFilterBoxBorderAdvanced_32f_C1R_Ctx (C function), [970](#)
 nppiFilterBoxBorderAdvanced_32f_C3R_Ctx (C function), [970](#)
 nppiFilterBoxBorderAdvanced_32f_C4R_Ctx (C function), [970](#)
 nppiFilterBoxBorderAdvanced_64f_C1R_Ctx (C function), [971](#)
 nppiFilterBoxBorderAdvancedGetDeviceBufferSize (C function), [967](#)
 nppiFilterBoxBorderAdvancedGetDeviceBufferSize_64 (C function), [968](#)
 nppiFilterCannyBorder_8u_C1R_Ctx (C function), [1167](#)
 nppiFilterCannyBorder_8u_C3C1R_Ctx (C function), [1168](#)
 nppiFilterCannyBorderGetBufferSize (C function), [1167](#)
 nppiFilterColumn32f_8u_AC4R_Ctx (C function), [965](#)

tion), 906
 nppiFilterColumn32f_8u_C1R_Ctx (C function), 905
 nppiFilterColumn32f_8u_C3R_Ctx (C function), 905
 nppiFilterColumn32f_8u_C4R_Ctx (C function), 906
 nppiFilterColumn32f_16s_AC4R_Ctx (C function), 907
 nppiFilterColumn32f_16s_C1R_Ctx (C function), 907
 nppiFilterColumn32f_16s_C3R_Ctx (C function), 907
 nppiFilterColumn32f_16s_C4R_Ctx (C function), 907
 nppiFilterColumn32f_16u_AC4R_Ctx (C function), 906
 nppiFilterColumn32f_16u_C1R_Ctx (C function), 906
 nppiFilterColumn32f_16u_C3R_Ctx (C function), 906
 nppiFilterColumn32f_16u_C4R_Ctx (C function), 906
 nppiFilterColumn_8u_AC4R_Ctx (C function), 897
 nppiFilterColumn_8u_C1R_Ctx (C function), 897
 nppiFilterColumn_8u_C3R_Ctx (C function), 897
 nppiFilterColumn_8u_C4R_Ctx (C function), 897
 nppiFilterColumn_16s_AC4R_Ctx (C function), 899
 nppiFilterColumn_16s_C1R_Ctx (C function), 898
 nppiFilterColumn_16s_C3R_Ctx (C function), 898
 nppiFilterColumn_16s_C4R_Ctx (C function), 899
 nppiFilterColumn_16u_AC4R_Ctx (C function), 898
 nppiFilterColumn_16u_C1R_Ctx (C function), 898
 nppiFilterColumn_16u_C3R_Ctx (C function), 898
 nppiFilterColumn_16u_C4R_Ctx (C function), 898
 nppiFilterColumn_32f_AC4R_Ctx (C function), 899
 nppiFilterColumn_32f_C1R_Ctx (C function), 899
 nppiFilterColumn_32f_C3R_Ctx (C function), 899
 nppiFilterColumn_32f_C4R_Ctx (C function), 899
 nppiFilterColumn_64f_C1R_Ctx (C function), 899
 nppiFilterColumnBorder32f_8u_AC4R_Ctx (C function), 909
 nppiFilterColumnBorder32f_8u_C1R_Ctx (C function), 908
 nppiFilterColumnBorder32f_8u_C3R_Ctx (C function), 908
 nppiFilterColumnBorder32f_8u_C4R_Ctx (C function), 909
 nppiFilterColumnBorder32f_16s_AC4R_Ctx (C function), 911
 nppiFilterColumnBorder32f_16s_C1R_Ctx (C function), 910
 nppiFilterColumnBorder32f_16s_C3R_Ctx (C function), 910
 nppiFilterColumnBorder32f_16s_C4R_Ctx (C function), 910
 nppiFilterColumnBorder32f_16u_AC4R_Ctx (C function), 910
 nppiFilterColumnBorder32f_16u_C1R_Ctx (C function), 909
 nppiFilterColumnBorder32f_16u_C3R_Ctx (C function), 909
 nppiFilterColumnBorder32f_16u_C4R_Ctx (C function), 910
 nppiFilterColumnBorder_8u_AC4R_Ctx (C function), 901
 nppiFilterColumnBorder_8u_C1R_Ctx (C function), 901
 nppiFilterColumnBorder_8u_C3R_Ctx (C function), 901
 nppiFilterColumnBorder_8u_C4R_Ctx (C function), 901
 nppiFilterColumnBorder_16s_AC4R_Ctx (C function), 903
 nppiFilterColumnBorder_16s_C1R_Ctx (C function), 902
 nppiFilterColumnBorder_16s_C3R_Ctx (C function), 903
 nppiFilterColumnBorder_16s_C4R_Ctx (C function), 903
 nppiFilterColumnBorder_16u_AC4R_Ctx (C function), 902
 nppiFilterColumnBorder_16u_C1R_Ctx (C function), 902
 nppiFilterColumnBorder_16u_C3R_Ctx (C function), 902
 nppiFilterColumnBorder_16u_C4R_Ctx (C function), 902
 nppiFilterColumnBorder_32f_AC4R_Ctx (C function), 904
 nppiFilterColumnBorder_32f_C1R_Ctx (C

<i>function</i>), 903	<i>function</i>), 1053
nppiFilterColumnBorder_32f_C3R_Ctx (C <i>function</i>), 904	nppiFilterGaussAdvanced_16u_C1R_Ctx (C <i>function</i>), 1052
nppiFilterColumnBorder_32f_C4R_Ctx (C <i>function</i>), 904	nppiFilterGaussAdvanced_16u_C3R_Ctx (C <i>function</i>), 1052
nppiFilterGauss_8u_AC4R_Ctx (C <i>function</i>), 1049	nppiFilterGaussAdvanced_16u_C4R_Ctx (C <i>function</i>), 1053
nppiFilterGauss_8u_C1R_Ctx (C <i>function</i>), 1049	nppiFilterGaussAdvanced_32f_AC4R_Ctx (C <i>function</i>), 1054
nppiFilterGauss_8u_C3R_Ctx (C <i>function</i>), 1049	nppiFilterGaussAdvanced_32f_C1R_Ctx (C <i>function</i>), 1054
nppiFilterGauss_8u_C4R_Ctx (C <i>function</i>), 1049	nppiFilterGaussAdvanced_32f_C3R_Ctx (C <i>function</i>), 1054
nppiFilterGauss_16s_AC4R_Ctx (C <i>function</i>), 1050	nppiFilterGaussAdvanced_32f_C4R_Ctx (C <i>function</i>), 1054
nppiFilterGauss_16s_C1R_Ctx (C <i>function</i>), 1050	nppiFilterGaussAdvancedBorder_8u_AC4R_Ctx (C <i>function</i>), 1060
nppiFilterGauss_16s_C3R_Ctx (C <i>function</i>), 1050	nppiFilterGaussAdvancedBorder_8u_C1R_Ctx (C <i>function</i>), 1060
nppiFilterGauss_16s_C4R_Ctx (C <i>function</i>), 1050	nppiFilterGaussAdvancedBorder_8u_C3R_Ctx (C <i>function</i>), 1060
nppiFilterGauss_16u_AC4R_Ctx (C <i>function</i>), 1050	nppiFilterGaussAdvancedBorder_8u_C4R_Ctx (C <i>function</i>), 1060
nppiFilterGauss_16u_C1R_Ctx (C <i>function</i>), 1049	nppiFilterGaussAdvancedBorder_16s_AC4R_Ctx (C <i>function</i>), 1062
nppiFilterGauss_16u_C3R_Ctx (C <i>function</i>), 1050	nppiFilterGaussAdvancedBorder_16s_C1R_Ctx (C <i>function</i>), 1062
nppiFilterGauss_16u_C4R_Ctx (C <i>function</i>), 1050	nppiFilterGaussAdvancedBorder_16s_C3R_Ctx (C <i>function</i>), 1062
nppiFilterGauss_32f_AC4R_Ctx (C <i>function</i>), 1051	nppiFilterGaussAdvancedBorder_16s_C4R_Ctx (C <i>function</i>), 1062
nppiFilterGauss_32f_C1R_Ctx (C <i>function</i>), 1050	nppiFilterGaussAdvancedBorder_16u_AC4R_Ctx (C <i>function</i>), 1061
nppiFilterGauss_32f_C3R_Ctx (C <i>function</i>), 1051	nppiFilterGaussAdvancedBorder_16u_C1R_Ctx (C <i>function</i>), 1061
nppiFilterGauss_32f_C4R_Ctx (C <i>function</i>), 1051	nppiFilterGaussAdvancedBorder_16u_C3R_Ctx (C <i>function</i>), 1061
nppiFilterGaussAdvanced_8u_AC4R_Ctx (C <i>function</i>), 1052	nppiFilterGaussAdvancedBorder_16u_C4R_Ctx (C <i>function</i>), 1061
nppiFilterGaussAdvanced_8u_C1R_Ctx (C <i>function</i>), 1052	nppiFilterGaussAdvancedBorder_32f_AC4R_Ctx (C <i>function</i>), 1063
nppiFilterGaussAdvanced_8u_C3R_Ctx (C <i>function</i>), 1052	nppiFilterGaussAdvancedBorder_32f_C1R_Ctx (C <i>function</i>), 1063
nppiFilterGaussAdvanced_8u_C4R_Ctx (C <i>function</i>), 1052	nppiFilterGaussAdvancedBorder_32f_C3R_Ctx (C <i>function</i>), 1063
nppiFilterGaussAdvanced_16s_AC4R_Ctx (C <i>function</i>), 1054	nppiFilterGaussAdvancedBorder_32f_C4R_Ctx (C <i>function</i>), 1063
nppiFilterGaussAdvanced_16s_C1R_Ctx (C <i>function</i>), 1053	nppiFilterGaussBorder_8u_AC4R_Ctx (C <i>function</i>), 1056
nppiFilterGaussAdvanced_16s_C3R_Ctx (C <i>function</i>), 1053	nppiFilterGaussBorder_8u_C1R_Ctx (C <i>function</i>), 1056
nppiFilterGaussAdvanced_16s_C4R_Ctx (C <i>function</i>), 1053	nppiFilterGaussBorder_8u_C3R_Ctx (C <i>function</i>), 1056
nppiFilterGaussAdvanced_16u_AC4R_Ctx (C	nppiFilterGaussBorder_8u_C4R_Ctx (C <i>func-</i>

tion), 1056
 nppiFilterGaussBorder_16s_AC4R_Ctx (C function), 1058
 nppiFilterGaussBorder_16s_C1R_Ctx (C function), 1057
 nppiFilterGaussBorder_16s_C3R_Ctx (C function), 1057
 nppiFilterGaussBorder_16s_C4R_Ctx (C function), 1057
 nppiFilterGaussBorder_16u_AC4R_Ctx (C function), 1057
 nppiFilterGaussBorder_16u_C1R_Ctx (C function), 1056
 nppiFilterGaussBorder_16u_C3R_Ctx (C function), 1056
 nppiFilterGaussBorder_16u_C4R_Ctx (C function), 1057
 nppiFilterGaussBorder_32f_AC4R_Ctx (C function), 1058
 nppiFilterGaussBorder_32f_C1R_Ctx (C function), 1058
 nppiFilterGaussBorder_32f_C3R_Ctx (C function), 1058
 nppiFilterGaussBorder_32f_C4R_Ctx (C function), 1058
 nppiFilterGaussPyramidLayerDownBorder_8u_AC4R_Ctx (C function), 1065
 nppiFilterGaussPyramidLayerDownBorder_8u_C1R_Ctx (C function), 1066
 nppiFilterGaussPyramidLayerDownBorder_8u_C3R_Ctx (C function), 1066
 nppiFilterGaussPyramidLayerDownBorder_8u_C4R_Ctx (C function), 1066
 nppiFilterGaussPyramidLayerDownBorder_16u_AC4R_Ctx (C function), 1066
 nppiFilterGaussPyramidLayerDownBorder_16u_C1R_Ctx (C function), 1067
 nppiFilterGaussPyramidLayerDownBorder_16u_C3R_Ctx (C function), 1067
 nppiFilterGaussPyramidLayerDownBorder_16u_C4R_Ctx (C function), 1067
 nppiFilterGaussPyramidLayerUpBorder_8u_AC4R_Ctx (C function), 1069
 nppiFilterGaussPyramidLayerUpBorder_8u_C1R_Ctx (C function), 1069
 nppiFilterGaussPyramidLayerUpBorder_8u_C3R_Ctx (C function), 1069
 nppiFilterGaussPyramidLayerUpBorder_8u_C4R_Ctx (C function), 1069
 nppiFilterGaussPyramidLayerUpBorder_16u_AC4R_Ctx (C function), 1069
 nppiFilterGaussPyramidLayerUpBorder_16u_C1R_Ctx (C function), 1070
 nppiFilterGaussPyramidLayerUpBorder_16u_C3R_Ctx (C function), 1070
 nppiFilterGaussPyramidLayerUpBorder_16u_C4R_Ctx (C function), 1070
 nppiFilterGaussPyramidLayerUpBorder_32f_AC4R_Ctx (C function), 1070
 nppiFilterGaussPyramidLayerUpBorder_32f_C1R_Ctx (C function), 1070
 nppiFilterGaussPyramidLayerUpBorder_32f_C3R_Ctx (C function), 1070
 nppiFilterGaussPyramidLayerUpBorder_32f_C4R_Ctx (C function), 1070
 nppiFilterHarrisCornersBorder_8u32f_C1R_Ctx (C function), 1170
 nppiFilterHarrisCornersBorderGetBufferSize (C function), 1169
 nppiFilterHighPass_8u_AC4R_Ctx (C function), 1076
 nppiFilterHighPass_8u_C1R_Ctx (C function), 1075
 nppiFilterHighPass_8u_C3R_Ctx (C function), 1075
 nppiFilterHighPass_8u_C4R_Ctx (C function), 1076
 nppiFilterHighPass_16s_AC4R_Ctx (C function), 1077
 nppiFilterHighPass_16s_C1R_Ctx (C function), 1076
 nppiFilterHighPass_16s_C3R_Ctx (C function), 1077
 nppiFilterHighPass_16s_C4R_Ctx (C function), 1077
 nppiFilterHighPass_16u_AC4R_Ctx (C function), 1076
 nppiFilterHighPass_16u_C1R_Ctx (C function), 1076
 nppiFilterHighPass_16u_C3R_Ctx (C function), 1076
 nppiFilterHighPass_16u_C4R_Ctx (C function), 1076
 nppiFilterHighPass_32f_AC4R_Ctx (C function), 1077
 nppiFilterHighPass_32f_C1R_Ctx (C function), 1077
 nppiFilterHighPass_32f_C3R_Ctx (C function), 1077
 nppiFilterHighPass_32f_C4R_Ctx (C function), 1077
 nppiFilterHighPassBorder_8u_AC4R_Ctx (C function), 1079
 nppiFilterHighPassBorder_8u_C1R_Ctx (C function), 1079
 nppiFilterHighPassBorder_8u_C3R_Ctx (C function), 1079
 nppiFilterHighPassBorder_8u_C4R_Ctx (C function), 1079
 nppiFilterHighPassBorder_16s_AC4R_Ctx (C function), 1081
 nppiFilterHighPassBorder_16s_C1R_Ctx (C function), 1080
 nppiFilterHighPassBorder_16s_C3R_Ctx (C function), 1080
 nppiFilterHighPassBorder_16s_C4R_Ctx (C function), 1080
 nppiFilterHighPassBorder_16u_AC4R_Ctx (C function), 1080
 nppiFilterHighPassBorder_16u_C1R_Ctx (C function), 1079
 nppiFilterHighPassBorder_16u_C3R_Ctx (C function), 1079
 nppiFilterHighPassBorder_16u_C4R_Ctx (C function), 1079

function), 1080
 nppiFilterHighPassBorder_32f_AC4R_Ctx (C function), 1081
 nppiFilterHighPassBorder_32f_C1R_Ctx (C function), 1081
 nppiFilterHighPassBorder_32f_C3R_Ctx (C function), 1081
 nppiFilterHighPassBorder_32f_C4R_Ctx (C function), 1081
 nppiFilterHoughLine_8u32f_C1R_Ctx (C function), 1172
 nppiFilterHoughLineGetBufferSize (C function), 1172
 nppiFilterHoughLineRegion_8u32f_C1R_Ctx (C function), 1173
 nppiFilterLaplace_8s16s_C1R_Ctx (C function), 1044
 nppiFilterLaplace_8u16s_C1R_Ctx (C function), 1044
 nppiFilterLaplace_8u_AC4R_Ctx (C function), 1043
 nppiFilterLaplace_8u_C1R_Ctx (C function), 1042
 nppiFilterLaplace_8u_C3R_Ctx (C function), 1042
 nppiFilterLaplace_8u_C4R_Ctx (C function), 1043
 nppiFilterLaplace_16s_AC4R_Ctx (C function), 1043
 nppiFilterLaplace_16s_C1R_Ctx (C function), 1043
 nppiFilterLaplace_16s_C3R_Ctx (C function), 1043
 nppiFilterLaplace_16s_C4R_Ctx (C function), 1043
 nppiFilterLaplace_32f_AC4R_Ctx (C function), 1044
 nppiFilterLaplace_32f_C1R_Ctx (C function), 1043
 nppiFilterLaplace_32f_C3R_Ctx (C function), 1043
 nppiFilterLaplace_32f_C4R_Ctx (C function), 1044
 nppiFilterLaplaceBorder_8s16s_C1R_Ctx (C function), 1048
 nppiFilterLaplaceBorder_8u16s_C1R_Ctx (C function), 1048
 nppiFilterLaplaceBorder_8u_AC4R_Ctx (C function), 1046
 nppiFilterLaplaceBorder_8u_C1R_Ctx (C function), 1045
 nppiFilterLaplaceBorder_8u_C3R_Ctx (C function), 1045
 nppiFilterLaplaceBorder_8u_C4R_Ctx (C function), 1046
 nppiFilterLaplaceBorder_16s_AC4R_Ctx (C function), 1047
 nppiFilterLaplaceBorder_16s_C1R_Ctx (C function), 1046
 nppiFilterLaplaceBorder_16s_C3R_Ctx (C function), 1046
 nppiFilterLaplaceBorder_16s_C4R_Ctx (C function), 1046
 nppiFilterLaplaceBorder_32f_AC4R_Ctx (C function), 1047
 nppiFilterLaplaceBorder_32f_C1R_Ctx (C function), 1047
 nppiFilterLaplaceBorder_32f_C3R_Ctx (C function), 1047
 nppiFilterLaplaceBorder_32f_C4R_Ctx (C function), 1047
 nppiFilterLowPass_8u_AC4R_Ctx (C function), 1083
 nppiFilterLowPass_8u_C1R_Ctx (C function), 1083
 nppiFilterLowPass_8u_C3R_Ctx (C function), 1083
 nppiFilterLowPass_8u_C4R_Ctx (C function), 1083
 nppiFilterLowPass_16s_AC4R_Ctx (C function), 1084
 nppiFilterLowPass_16s_C1R_Ctx (C function), 1084
 nppiFilterLowPass_16s_C3R_Ctx (C function), 1084
 nppiFilterLowPass_16s_C4R_Ctx (C function), 1084
 nppiFilterLowPass_16u_AC4R_Ctx (C function), 1083
 nppiFilterLowPass_16u_C1R_Ctx (C function), 1083
 nppiFilterLowPass_16u_C3R_Ctx (C function), 1083
 nppiFilterLowPass_16u_C4R_Ctx (C function), 1083
 nppiFilterLowPass_32f_AC4R_Ctx (C function), 1085
 nppiFilterLowPass_32f_C1R_Ctx (C function), 1084
 nppiFilterLowPass_32f_C3R_Ctx (C function), 1084
 nppiFilterLowPass_32f_C4R_Ctx (C function), 1084
 nppiFilterLowPassBorder_8u_AC4R_Ctx (C function), 1086
 nppiFilterLowPassBorder_8u_C1R_Ctx (C function), 1086
 nppiFilterLowPassBorder_8u_C3R_Ctx (C

function), 1086
 nppiFilterLowPassBorder_8u_C4R_Ctx (C *function*), 1086
 nppiFilterLowPassBorder_16s_AC4R_Ctx (C *function*), 1088
 nppiFilterLowPassBorder_16s_C1R_Ctx (C *function*), 1087
 nppiFilterLowPassBorder_16s_C3R_Ctx (C *function*), 1087
 nppiFilterLowPassBorder_16s_C4R_Ctx (C *function*), 1088
 nppiFilterLowPassBorder_16u_AC4R_Ctx (C *function*), 1087
 nppiFilterLowPassBorder_16u_C1R_Ctx (C *function*), 1086
 nppiFilterLowPassBorder_16u_C3R_Ctx (C *function*), 1087
 nppiFilterLowPassBorder_16u_C4R_Ctx (C *function*), 1087
 nppiFilterLowPassBorder_32f_AC4R_Ctx (C *function*), 1089
 nppiFilterLowPassBorder_32f_C1R_Ctx (C *function*), 1088
 nppiFilterLowPassBorder_32f_C3R_Ctx (C *function*), 1088
 nppiFilterLowPassBorder_32f_C4R_Ctx (C *function*), 1088
 nppiFilterMax_8u_AC4R_Ctx (C *function*), 974
 nppiFilterMax_8u_C1R_Ctx (C *function*), 973
 nppiFilterMax_8u_C3R_Ctx (C *function*), 973
 nppiFilterMax_8u_C4R_Ctx (C *function*), 973
 nppiFilterMax_16s_AC4R_Ctx (C *function*), 975
 nppiFilterMax_16s_C1R_Ctx (C *function*), 974
 nppiFilterMax_16s_C3R_Ctx (C *function*), 974
 nppiFilterMax_16s_C4R_Ctx (C *function*), 975
 nppiFilterMax_16u_AC4R_Ctx (C *function*), 974
 nppiFilterMax_16u_C1R_Ctx (C *function*), 974
 nppiFilterMax_16u_C3R_Ctx (C *function*), 974
 nppiFilterMax_16u_C4R_Ctx (C *function*), 974
 nppiFilterMax_32f_AC4R_Ctx (C *function*), 975
 nppiFilterMax_32f_C1R_Ctx (C *function*), 975
 nppiFilterMax_32f_C3R_Ctx (C *function*), 975
 nppiFilterMax_32f_C4R_Ctx (C *function*), 975
 nppiFilterMaxBorder_8u_AC4R_Ctx (C *function*), 977
 nppiFilterMaxBorder_8u_C1R_Ctx (C *function*), 977
 nppiFilterMaxBorder_8u_C3R_Ctx (C *function*), 977
 nppiFilterMaxBorder_8u_C4R_Ctx (C *function*), 977
 nppiFilterMaxBorder_16s_AC4R_Ctx (C *function*), 978
 nppiFilterMaxBorder_16s_C1R_Ctx (C *function*), 978
 nppiFilterMaxBorder_16s_C3R_Ctx (C *function*), 978
 nppiFilterMaxBorder_16s_C4R_Ctx (C *function*), 978
 nppiFilterMaxBorder_16u_AC4R_Ctx (C *function*), 978
 nppiFilterMaxBorder_16u_C1R_Ctx (C *function*), 977
 nppiFilterMaxBorder_16u_C3R_Ctx (C *function*), 977
 nppiFilterMaxBorder_16u_C4R_Ctx (C *function*), 978
 nppiFilterMaxBorder_32f_AC4R_Ctx (C *function*), 979
 nppiFilterMaxBorder_32f_C1R_Ctx (C *function*), 979
 nppiFilterMaxBorder_32f_C3R_Ctx (C *function*), 979
 nppiFilterMaxBorder_32f_C4R_Ctx (C *function*), 979
 nppiFilterMedian_8u_AC4R_Ctx (C *function*), 987
 nppiFilterMedian_8u_C1R_Ctx (C *function*), 987
 nppiFilterMedian_8u_C3R_Ctx (C *function*), 987
 nppiFilterMedian_8u_C4R_Ctx (C *function*), 987
 nppiFilterMedian_16s_AC4R_Ctx (C *function*), 988
 nppiFilterMedian_16s_C1R_Ctx (C *function*), 988
 nppiFilterMedian_16s_C3R_Ctx (C *function*), 988
 nppiFilterMedian_16s_C4R_Ctx (C *function*), 988
 nppiFilterMedian_16u_AC4R_Ctx (C *function*), 988
 nppiFilterMedian_16u_C1R_Ctx (C *function*), 987
 nppiFilterMedian_16u_C3R_Ctx (C *function*), 988
 nppiFilterMedian_16u_C4R_Ctx (C *function*), 988
 nppiFilterMedian_32f_AC4R_Ctx (C *function*), 989
 nppiFilterMedian_32f_C1R_Ctx (C *function*), 989
 nppiFilterMedian_32f_C3R_Ctx (C *function*), 989

nppiFilterMedian_32f_C4R_Ctx (C function), 989
 nppiFilterMedianBorder_8u_AC4R_Ctx (C function), 993
 nppiFilterMedianBorder_8u_C1R_Ctx (C function), 993
 nppiFilterMedianBorder_8u_C3R_Ctx (C function), 993
 nppiFilterMedianBorder_8u_C4R_Ctx (C function), 993
 nppiFilterMedianBorder_16s_AC4R_Ctx (C function), 995
 nppiFilterMedianBorder_16s_C1R_Ctx (C function), 994
 nppiFilterMedianBorder_16s_C3R_Ctx (C function), 994
 nppiFilterMedianBorder_16s_C4R_Ctx (C function), 994
 nppiFilterMedianBorder_16u_AC4R_Ctx (C function), 994
 nppiFilterMedianBorder_16u_C1R_Ctx (C function), 993
 nppiFilterMedianBorder_16u_C3R_Ctx (C function), 993
 nppiFilterMedianBorder_16u_C4R_Ctx (C function), 994
 nppiFilterMedianBorder_32f_AC4R_Ctx (C function), 995
 nppiFilterMedianBorder_32f_C1R_Ctx (C function), 995
 nppiFilterMedianBorder_32f_C3R_Ctx (C function), 995
 nppiFilterMedianBorder_32f_C4R_Ctx (C function), 995
 nppiFilterMedianBorderGetBufferSize_8u_AC4R_Ctx (C function), 996
 nppiFilterMedianBorderGetBufferSize_8u_C1R_Ctx (C function), 996
 nppiFilterMedianBorderGetBufferSize_8u_C3R_Ctx (C function), 996
 nppiFilterMedianBorderGetBufferSize_8u_C4R_Ctx (C function), 996
 nppiFilterMedianBorderGetBufferSize_16s_AC4R_Ctx (C function), 998
 nppiFilterMedianBorderGetBufferSize_16s_C1R_Ctx (C function), 997
 nppiFilterMedianBorderGetBufferSize_16s_C3R_Ctx (C function), 998
 nppiFilterMedianBorderGetBufferSize_16s_C4R_Ctx (C function), 998
 nppiFilterMedianBorderGetBufferSize_16u_AC4R_Ctx (C function), 997
 nppiFilterMedianBorderGetBufferSize_16u_C1R_Ctx (C function), 997
 nppiFilterMedianBorderGetBufferSize_16u_C3R_Ctx (C function), 997
 nppiFilterMedianBorderGetBufferSize_16u_C4R_Ctx (C function), 997
 nppiFilterMedianBorderGetBufferSize_32f_AC4R_Ctx (C function), 999
 nppiFilterMedianBorderGetBufferSize_32f_C1R_Ctx (C function), 998
 nppiFilterMedianBorderGetBufferSize_32f_C3R_Ctx (C function), 998
 nppiFilterMedianBorderGetBufferSize_32f_C4R_Ctx (C function), 999
 nppiFilterMedianGetBufferSize_8u_AC4R_Ctx (C function), 990
 nppiFilterMedianGetBufferSize_8u_C1R_Ctx (C function), 989
 nppiFilterMedianGetBufferSize_8u_C3R_Ctx (C function), 989
 nppiFilterMedianGetBufferSize_8u_C4R_Ctx (C function), 989
 nppiFilterMedianGetBufferSize_16s_AC4R_Ctx (C function), 991
 nppiFilterMedianGetBufferSize_16s_C1R_Ctx (C function), 990
 nppiFilterMedianGetBufferSize_16s_C3R_Ctx (C function), 990
 nppiFilterMedianGetBufferSize_16s_C4R_Ctx (C function), 990
 nppiFilterMedianGetBufferSize_16u_AC4R_Ctx (C function), 990
 nppiFilterMedianGetBufferSize_16u_C1R_Ctx (C function), 990
 nppiFilterMedianGetBufferSize_16u_C3R_Ctx (C function), 990
 nppiFilterMedianGetBufferSize_16u_C4R_Ctx (C function), 990
 nppiFilterMedianGetBufferSize_32f_AC4R_Ctx (C function), 991
 nppiFilterMedianGetBufferSize_32f_C1R_Ctx (C function), 991
 nppiFilterMedianGetBufferSize_32f_C3R_Ctx (C function), 991
 nppiFilterMedianGetBufferSize_32f_C4R_Ctx (C function), 991
 nppiFilterMin_8u_AC4R_Ctx (C function), 980
 nppiFilterMin_8u_C1R_Ctx (C function), 980
 nppiFilterMin_8u_C3R_Ctx (C function), 980
 nppiFilterMin_8u_C4R_Ctx (C function), 980
 nppiFilterMin_16s_AC4R_Ctx (C function), 982
 nppiFilterMin_16s_C1R_Ctx (C function), 981
 nppiFilterMin_16s_C3R_Ctx (C function), 981
 nppiFilterMin_16s_C4R_Ctx (C function), 981
 nppiFilterMin_16u_AC4R_Ctx (C function),

981
 nppiFilterMin_16u_C1R_Ctx (C function), 981
 nppiFilterMin_16u_C3R_Ctx (C function), 981
 nppiFilterMin_16u_C4R_Ctx (C function), 981
 nppiFilterMin_32f_AC4R_Ctx (C function), 982
 nppiFilterMin_32f_C1R_Ctx (C function), 982
 nppiFilterMin_32f_C3R_Ctx (C function), 982
 nppiFilterMin_32f_C4R_Ctx (C function), 982
 nppiFilterMinBorder_8u_AC4R_Ctx (C function), 984
 nppiFilterMinBorder_8u_C1R_Ctx (C function), 983
 nppiFilterMinBorder_8u_C3R_Ctx (C function), 983
 nppiFilterMinBorder_8u_C4R_Ctx (C function), 983
 nppiFilterMinBorder_16s_AC4R_Ctx (C function), 985
 nppiFilterMinBorder_16s_C1R_Ctx (C function), 985
 nppiFilterMinBorder_16s_C3R_Ctx (C function), 985
 nppiFilterMinBorder_16s_C4R_Ctx (C function), 985
 nppiFilterMinBorder_16u_AC4R_Ctx (C function), 984
 nppiFilterMinBorder_16u_C1R_Ctx (C function), 984
 nppiFilterMinBorder_16u_C3R_Ctx (C function), 984
 nppiFilterMinBorder_16u_C4R_Ctx (C function), 984
 nppiFilterMinBorder_32f_AC4R_Ctx (C function), 986
 nppiFilterMinBorder_32f_C1R_Ctx (C function), 985
 nppiFilterMinBorder_32f_C3R_Ctx (C function), 985
 nppiFilterMinBorder_32f_C4R_Ctx (C function), 986
 nppiFilterPrewittHoriz_8u_AC4R_Ctx (C function), 1001
 nppiFilterPrewittHoriz_8u_C1R_Ctx (C function), 1000
 nppiFilterPrewittHoriz_8u_C3R_Ctx (C function), 1000
 nppiFilterPrewittHoriz_8u_C4R_Ctx (C function), 1000
 nppiFilterPrewittHoriz_16s_AC4R_Ctx (C function), 1001
 nppiFilterPrewittHoriz_16s_C1R_Ctx (C function), 1001
 nppiFilterPrewittHoriz_16s_C3R_Ctx (C function), 1001
 nppiFilterPrewittHoriz_16s_C4R_Ctx (C function), 1001
 nppiFilterPrewittHoriz_32f_AC4R_Ctx (C function), 1002
 nppiFilterPrewittHoriz_32f_C1R_Ctx (C function), 1001
 nppiFilterPrewittHoriz_32f_C3R_Ctx (C function), 1001
 nppiFilterPrewittHoriz_32f_C4R_Ctx (C function), 1002
 nppiFilterPrewittHorizBorder_8u_AC4R_Ctx (C function), 1005
 nppiFilterPrewittHorizBorder_8u_C1R_Ctx (C function), 1005
 nppiFilterPrewittHorizBorder_8u_C3R_Ctx (C function), 1005
 nppiFilterPrewittHorizBorder_8u_C4R_Ctx (C function), 1005
 nppiFilterPrewittHorizBorder_16s_AC4R_Ctx (C function), 1006
 nppiFilterPrewittHorizBorder_16s_C1R_Ctx (C function), 1005
 nppiFilterPrewittHorizBorder_16s_C3R_Ctx (C function), 1006
 nppiFilterPrewittHorizBorder_16s_C4R_Ctx (C function), 1006
 nppiFilterPrewittHorizBorder_32f_AC4R_Ctx (C function), 1007
 nppiFilterPrewittHorizBorder_32f_C1R_Ctx (C function), 1006
 nppiFilterPrewittHorizBorder_32f_C3R_Ctx (C function), 1006
 nppiFilterPrewittHorizBorder_32f_C4R_Ctx (C function), 1007
 nppiFilterPrewittVert_8u_AC4R_Ctx (C function), 1002
 nppiFilterPrewittVert_8u_C1R_Ctx (C function), 1002
 nppiFilterPrewittVert_8u_C3R_Ctx (C function), 1002
 nppiFilterPrewittVert_8u_C4R_Ctx (C function), 1002
 nppiFilterPrewittVert_16s_AC4R_Ctx (C function), 1003
 nppiFilterPrewittVert_16s_C1R_Ctx (C function), 1003
 nppiFilterPrewittVert_16s_C3R_Ctx (C function), 1003
 nppiFilterPrewittVert_16s_C4R_Ctx (C function), 1003
 nppiFilterPrewittVert_32f_AC4R_Ctx (C function), 1003
 nppiFilterPrewittVert_32f_C1R_Ctx (C function), 1003

<i>function</i>), 1003		<i>(C function)</i> , 1037
nppiFilterPrewittVert_32f_C3R_Ctx	(C	nppiFilterRobertsDownBorder_8u_C1R_Ctx
<i>function</i>), 1003		<i>(C function)</i> , 1036
nppiFilterPrewittVert_32f_C4R_Ctx	(C	nppiFilterRobertsDownBorder_8u_C3R_Ctx
<i>function</i>), 1003		<i>(C function)</i> , 1037
nppiFilterPrewittVertBorder_8u_AC4R_Ctx		nppiFilterRobertsDownBorder_8u_C4R_Ctx
<i>(C function)</i> , 1008		<i>(C function)</i> , 1037
nppiFilterPrewittVertBorder_8u_C1R_Ctx		nppiFilterRobertsDownBorder_16s_AC4R_Ctx
<i>(C function)</i> , 1007		<i>(C function)</i> , 1038
nppiFilterPrewittVertBorder_8u_C3R_Ctx		nppiFilterRobertsDownBorder_16s_C1R_Ctx
<i>(C function)</i> , 1007		<i>(C function)</i> , 1037
nppiFilterPrewittVertBorder_8u_C4R_Ctx		nppiFilterRobertsDownBorder_16s_C3R_Ctx
<i>(C function)</i> , 1008		<i>(C function)</i> , 1037
nppiFilterPrewittVertBorder_16s_AC4R_Ctx	nppiFilterRobertsDownBorder_16s_C4R_Ctx	
<i>(C function)</i> , 1009	<i>(C function)</i> , 1038	
nppiFilterPrewittVertBorder_16s_C1R_Ctx	nppiFilterRobertsDownBorder_32f_AC4R_Ctx	
<i>(C function)</i> , 1008	<i>(C function)</i> , 1039	
nppiFilterPrewittVertBorder_16s_C3R_Ctx	nppiFilterRobertsDownBorder_32f_C1R_Ctx	
<i>(C function)</i> , 1008	<i>(C function)</i> , 1038	
nppiFilterPrewittVertBorder_16s_C4R_Ctx	nppiFilterRobertsDownBorder_32f_C3R_Ctx	
<i>(C function)</i> , 1008	<i>(C function)</i> , 1038	
nppiFilterPrewittVertBorder_32f_AC4R_Ctx	nppiFilterRobertsDownBorder_32f_C4R_Ctx	
<i>(C function)</i> , 1009	<i>(C function)</i> , 1038	
nppiFilterPrewittVertBorder_32f_C1R_Ctx	nppiFilterRobertsUp_8u_AC4R_Ctx	<i>(C func-</i>
<i>tion)</i> , 1009	<i>tion)</i> , 1034	
nppiFilterPrewittVertBorder_32f_C3R_Ctx	nppiFilterRobertsUp_8u_C1R_Ctx	<i>(C func-</i>
<i>tion)</i> , 1009	<i>tion)</i> , 1034	
nppiFilterPrewittVertBorder_32f_C4R_Ctx	nppiFilterRobertsUp_8u_C3R_Ctx	<i>(C func-</i>
<i>tion)</i> , 1009	<i>tion)</i> , 1034	
nppiFilterRobertsDown_8u_AC4R_Ctx	nppiFilterRobertsUp_8u_C4R_Ctx	<i>(C func-</i>
<i>function)</i> , 1033	<i>tion)</i> , 1034	
nppiFilterRobertsDown_8u_C1R_Ctx	nppiFilterRobertsUp_16s_AC4R_Ctx	<i>(C func-</i>
<i>tion)</i> , 1032	<i>tion)</i> , 1035	
nppiFilterRobertsDown_8u_C3R_Ctx	nppiFilterRobertsUp_16s_C1R_Ctx	<i>(C func-</i>
<i>tion)</i> , 1032	<i>tion)</i> , 1034	
nppiFilterRobertsDown_8u_C4R_Ctx	nppiFilterRobertsUp_16s_C3R_Ctx	<i>(C func-</i>
<i>tion)</i> , 1032	<i>tion)</i> , 1035	
nppiFilterRobertsDown_16s_AC4R_Ctx	nppiFilterRobertsUp_16s_C4R_Ctx	<i>(C func-</i>
<i>function)</i> , 1033	<i>tion)</i> , 1035	
nppiFilterRobertsDown_16s_C1R_Ctx	nppiFilterRobertsUp_32f_AC4R_Ctx	<i>(C func-</i>
<i>function)</i> , 1033	<i>tion)</i> , 1035	
nppiFilterRobertsDown_16s_C3R_Ctx	nppiFilterRobertsUp_32f_C1R_Ctx	<i>(C func-</i>
<i>function)</i> , 1033	<i>tion)</i> , 1035	
nppiFilterRobertsDown_16s_C4R_Ctx	nppiFilterRobertsUp_32f_C3R_Ctx	<i>(C func-</i>
<i>function)</i> , 1033	<i>tion)</i> , 1035	
nppiFilterRobertsDown_32f_AC4R_Ctx	nppiFilterRobertsUp_32f_C4R_Ctx	<i>(C func-</i>
<i>function)</i> , 1034	<i>tion)</i> , 1035	
nppiFilterRobertsDown_32f_C1R_Ctx	nppiFilterRobertsUpBorder_8u_AC4R_Ctx	
<i>function)</i> , 1033	<i>(C function)</i> , 1040	
nppiFilterRobertsDown_32f_C3R_Ctx	nppiFilterRobertsUpBorder_8u_C1R_Ctx	<i>(C</i>
<i>function)</i> , 1033	<i>function)</i> , 1039	
nppiFilterRobertsDown_32f_C4R_Ctx	nppiFilterRobertsUpBorder_8u_C3R_Ctx	<i>(C</i>
<i>function)</i> , 1033	<i>function)</i> , 1039	
nppiFilterRobertsDownBorder_8u_AC4R_Ctx	nppiFilterRobertsUpBorder_8u_C4R_Ctx	<i>(C</i>

function), 1040
 nppiFilterRobertsUpBorder_16s_AC4R_Ctx (C function), 1040
 nppiFilterRobertsUpBorder_16s_C1R_Ctx (C function), 1040
 nppiFilterRobertsUpBorder_16s_C3R_Ctx (C function), 1040
 nppiFilterRobertsUpBorder_16s_C4R_Ctx (C function), 1040
 nppiFilterRobertsUpBorder_32f_AC4R_Ctx (C function), 1041
 nppiFilterRobertsUpBorder_32f_C1R_Ctx (C function), 1041
 nppiFilterRobertsUpBorder_32f_C3R_Ctx (C function), 1041
 nppiFilterRobertsUpBorder_32f_C4R_Ctx (C function), 1041
 nppiFilterRow32f_8u_AC4R_Ctx (C function), 920
 nppiFilterRow32f_8u_C1R_Ctx (C function), 919
 nppiFilterRow32f_8u_C3R_Ctx (C function), 919
 nppiFilterRow32f_8u_C4R_Ctx (C function), 920
 nppiFilterRow32f_16s_AC4R_Ctx (C function), 921
 nppiFilterRow32f_16s_C1R_Ctx (C function), 920
 nppiFilterRow32f_16s_C3R_Ctx (C function), 921
 nppiFilterRow32f_16s_C4R_Ctx (C function), 921
 nppiFilterRow32f_16u_AC4R_Ctx (C function), 920
 nppiFilterRow32f_16u_C1R_Ctx (C function), 920
 nppiFilterRow32f_16u_C3R_Ctx (C function), 920
 nppiFilterRow32f_16u_C4R_Ctx (C function), 920
 nppiFilterRow_8u_AC4R_Ctx (C function), 912
 nppiFilterRow_8u_C1R_Ctx (C function), 912
 nppiFilterRow_8u_C3R_Ctx (C function), 912
 nppiFilterRow_8u_C4R_Ctx (C function), 912
 nppiFilterRow_16s_AC4R_Ctx (C function), 913
 nppiFilterRow_16s_C1R_Ctx (C function), 913
 nppiFilterRow_16s_C3R_Ctx (C function), 913
 nppiFilterRow_16s_C4R_Ctx (C function), 913
 nppiFilterRow_16u_AC4R_Ctx (C function), 913
 nppiFilterRow_16u_C1R_Ctx (C function), 912
 nppiFilterRow_16u_C3R_Ctx (C function), 913
 nppiFilterRow_16u_C4R_Ctx (C function), 913
 nppiFilterRow_16u_C4R_Ctx (C function), 913
 nppiFilterRow_32f_AC4R_Ctx (C function), 914
 nppiFilterRow_32f_C1R_Ctx (C function), 914
 nppiFilterRow_32f_C3R_Ctx (C function), 914
 nppiFilterRow_32f_C4R_Ctx (C function), 914
 nppiFilterRow_64f_C1R_Ctx (C function), 914
 nppiFilterRowBorder32f_8u_AC4R_Ctx (C function), 922
 nppiFilterRowBorder32f_8u_C1R_Ctx (C function), 922
 nppiFilterRowBorder32f_8u_C3R_Ctx (C function), 922
 nppiFilterRowBorder32f_8u_C4R_Ctx (C function), 922
 nppiFilterRowBorder32f_16s_AC4R_Ctx (C function), 924
 nppiFilterRowBorder32f_16s_C1R_Ctx (C function), 924
 nppiFilterRowBorder32f_16s_C3R_Ctx (C function), 924
 nppiFilterRowBorder32f_16s_C4R_Ctx (C function), 924
 nppiFilterRowBorder32f_16u_AC4R_Ctx (C function), 923
 nppiFilterRowBorder32f_16u_C1R_Ctx (C function), 923
 nppiFilterRowBorder32f_16u_C3R_Ctx (C function), 923
 nppiFilterRowBorder32f_16u_C4R_Ctx (C function), 923
 nppiFilterRowBorder_8u_AC4R_Ctx (C function), 916
 nppiFilterRowBorder_8u_C1R_Ctx (C function), 915
 nppiFilterRowBorder_8u_C3R_Ctx (C function), 915
 nppiFilterRowBorder_8u_C4R_Ctx (C function), 916
 nppiFilterRowBorder_16s_AC4R_Ctx (C function), 917
 nppiFilterRowBorder_16s_C1R_Ctx (C function), 917
 nppiFilterRowBorder_16s_C3R_Ctx (C function), 917
 nppiFilterRowBorder_16s_C4R_Ctx (C function), 917
 nppiFilterRowBorder_16u_AC4R_Ctx (C function), 917
 nppiFilterRowBorder_16u_C1R_Ctx (C function), 916
 nppiFilterRowBorder_16u_C3R_Ctx (C function), 916
 nppiFilterRowBorder_16u_C4R_Ctx (C function), 916

tion), 916
 nppiFilterRowBorder_32f_AC4R_Ctx (C function), 918
 nppiFilterRowBorder_32f_C1R_Ctx (C function), 918
 nppiFilterRowBorder_32f_C3R_Ctx (C function), 918
 nppiFilterRowBorder_32f_C4R_Ctx (C function), 918
 nppiFilterScharrHoriz_8s16s_C1R_Ctx (C function), 1011
 nppiFilterScharrHoriz_8u16s_C1R_Ctx (C function), 1010
 nppiFilterScharrHoriz_32f_C1R_Ctx (C function), 1011
 nppiFilterScharrHorizBorder_8s16s_C1R_Ctx (C function), 1013
 nppiFilterScharrHorizBorder_8u16s_C1R_Ctx (C function), 1012
 nppiFilterScharrHorizBorder_32f_C1R_Ctx (C function), 1013
 nppiFilterScharrVert_8s16s_C1R_Ctx (C function), 1011
 nppiFilterScharrVert_8u16s_C1R_Ctx (C function), 1011
 nppiFilterScharrVert_32f_C1R_Ctx (C function), 1011
 nppiFilterScharrVertBorder_8s16s_C1R_Ctx (C function), 1013
 nppiFilterScharrVertBorder_8u16s_C1R_Ctx (C function), 1013
 nppiFilterScharrVertBorder_32f_C1R_Ctx (C function), 1014
 nppiFilterSharpen_8u_AC4R_Ctx (C function), 1090
 nppiFilterSharpen_8u_C1R_Ctx (C function), 1090
 nppiFilterSharpen_8u_C3R_Ctx (C function), 1090
 nppiFilterSharpen_8u_C4R_Ctx (C function), 1090
 nppiFilterSharpen_16s_AC4R_Ctx (C function), 1091
 nppiFilterSharpen_16s_C1R_Ctx (C function), 1091
 nppiFilterSharpen_16s_C3R_Ctx (C function), 1091
 nppiFilterSharpen_16s_C4R_Ctx (C function), 1091
 nppiFilterSharpen_16u_AC4R_Ctx (C function), 1090
 nppiFilterSharpen_16u_C1R_Ctx (C function), 1090
 nppiFilterSharpen_16u_C3R_Ctx (C function), 1090
 nppiFilterSharpen_16u_C4R_Ctx (C function), 1090
 nppiFilterSharpen_16u_C1R_Ctx (C function), 1090
 nppiFilterSharpen_16u_C3R_Ctx (C function), 1090
 nppiFilterSharpen_16u_C4R_Ctx (C function), 1090
 nppiFilterSharpen_32f_AC4R_Ctx (C function), 1091
 nppiFilterSharpen_32f_C1R_Ctx (C function), 1091
 nppiFilterSharpen_32f_C3R_Ctx (C function), 1091
 nppiFilterSharpen_32f_C4R_Ctx (C function), 1091
 nppiFilterSharpenBorder_8u_AC4R_Ctx (C function), 1093
 nppiFilterSharpenBorder_8u_C1R_Ctx (C function), 1093
 nppiFilterSharpenBorder_8u_C3R_Ctx (C function), 1093
 nppiFilterSharpenBorder_8u_C4R_Ctx (C function), 1093
 nppiFilterSharpenBorder_16s_AC4R_Ctx (C function), 1094
 nppiFilterSharpenBorder_16s_C1R_Ctx (C function), 1094
 nppiFilterSharpenBorder_16s_C3R_Ctx (C function), 1094
 nppiFilterSharpenBorder_16s_C4R_Ctx (C function), 1094
 nppiFilterSharpenBorder_16u_AC4R_Ctx (C function), 1094
 nppiFilterSharpenBorder_16u_C1R_Ctx (C function), 1093
 nppiFilterSharpenBorder_16u_C3R_Ctx (C function), 1093
 nppiFilterSharpenBorder_16u_C4R_Ctx (C function), 1094
 nppiFilterSharpenBorder_32f_AC4R_Ctx (C function), 1095
 nppiFilterSharpenBorder_32f_C1R_Ctx (C function), 1095
 nppiFilterSharpenBorder_32f_C3R_Ctx (C function), 1095
 nppiFilterSharpenBorder_32f_C4R_Ctx (C function), 1095
 nppiFilterSobelCross_8s16s_C1R_Ctx (C function), 1021
 nppiFilterSobelCross_8u16s_C1R_Ctx (C function), 1020
 nppiFilterSobelCross_32f_C1R_Ctx (C function), 1021
 nppiFilterSobelCrossBorder_8s16s_C1R_Ctx (C function), 1031
 nppiFilterSobelCrossBorder_8u16s_C1R_Ctx (C function), 1031
 nppiFilterSobelCrossBorder_32f_C1R_Ctx (C function), 1031

(C function), 1031
 nppiFilterSobelHoriz_8s16s_C1R_Ctx (C function), 1016
 nppiFilterSobelHoriz_8u16s_C1R_Ctx (C function), 1016
 nppiFilterSobelHoriz_8u_AC4R_Ctx (C function), 1015
 nppiFilterSobelHoriz_8u_C1R_Ctx (C function), 1015
 nppiFilterSobelHoriz_8u_C3R_Ctx (C function), 1015
 nppiFilterSobelHoriz_8u_C4R_Ctx (C function), 1015
 nppiFilterSobelHoriz_16s_AC4R_Ctx (C function), 1016
 nppiFilterSobelHoriz_16s_C1R_Ctx (C function), 1015
 nppiFilterSobelHoriz_16s_C3R_Ctx (C function), 1015
 nppiFilterSobelHoriz_16s_C4R_Ctx (C function), 1015
 nppiFilterSobelHoriz_32f_AC4R_Ctx (C function), 1016
 nppiFilterSobelHoriz_32f_C1R_Ctx (C function), 1016
 nppiFilterSobelHoriz_32f_C3R_Ctx (C function), 1016
 nppiFilterSobelHoriz_32f_C4R_Ctx (C function), 1016
 nppiFilterSobelHorizBorder_8s16s_C1R_Ctx (C function), 1024
 nppiFilterSobelHorizBorder_8u16s_C1R_Ctx (C function), 1024
 nppiFilterSobelHorizBorder_8u_AC4R_Ctx (C function), 1022
 nppiFilterSobelHorizBorder_8u_C1R_Ctx (C function), 1022
 nppiFilterSobelHorizBorder_8u_C3R_Ctx (C function), 1022
 nppiFilterSobelHorizBorder_8u_C4R_Ctx (C function), 1022
 nppiFilterSobelHorizBorder_16s_AC4R_Ctx (C function), 1023
 nppiFilterSobelHorizBorder_16s_C1R_Ctx (C function), 1023
 nppiFilterSobelHorizBorder_16s_C3R_Ctx (C function), 1023
 nppiFilterSobelHorizBorder_16s_C4R_Ctx (C function), 1023
 nppiFilterSobelHorizBorder_32f_AC4R_Ctx (C function), 1024
 nppiFilterSobelHorizBorder_32f_C1R_Ctx (C function), 1023
 nppiFilterSobelHorizBorder_32f_C3R_Ctx (C function), 1024
 nppiFilterSobelHorizBorder_32f_C4R_Ctx (C function), 1024
 nppiFilterSobelHorizMask_32f_C1R_Ctx (C function), 1016
 nppiFilterSobelHorizMaskBorder_32f_C1R_Ctx (C function), 1025
 nppiFilterSobelHorizSecond_8s16s_C1R_Ctx (C function), 1019
 nppiFilterSobelHorizSecond_8u16s_C1R_Ctx (C function), 1019
 nppiFilterSobelHorizSecond_32f_C1R_Ctx (C function), 1019
 nppiFilterSobelHorizSecondBorder_8s16s_C1R_Ctx (C function), 1029
 nppiFilterSobelHorizSecondBorder_8u16s_C1R_Ctx (C function), 1028
 nppiFilterSobelHorizSecondBorder_32f_C1R_Ctx (C function), 1029
 nppiFilterSobelVert_8s16s_C1R_Ctx (C function), 1018
 nppiFilterSobelVert_8u16s_C1R_Ctx (C function), 1018
 nppiFilterSobelVert_8u_AC4R_Ctx (C function), 1017
 nppiFilterSobelVert_8u_C1R_Ctx (C function), 1017
 nppiFilterSobelVert_8u_C3R_Ctx (C function), 1017
 nppiFilterSobelVert_8u_C4R_Ctx (C function), 1017
 nppiFilterSobelVert_16s_AC4R_Ctx (C function), 1018
 nppiFilterSobelVert_16s_C1R_Ctx (C function), 1017
 nppiFilterSobelVert_16s_C3R_Ctx (C function), 1017
 nppiFilterSobelVert_16s_C4R_Ctx (C function), 1018
 nppiFilterSobelVert_32f_AC4R_Ctx (C function), 1018
 nppiFilterSobelVert_32f_C1R_Ctx (C function), 1018
 nppiFilterSobelVert_32f_C3R_Ctx (C function), 1018
 nppiFilterSobelVert_32f_C4R_Ctx (C function), 1018
 nppiFilterSobelVertBorder_8s16s_C1R_Ctx (C function), 1028
 nppiFilterSobelVertBorder_8u16s_C1R_Ctx (C function), 1027
 nppiFilterSobelVertBorder_8u_AC4R_Ctx (C function), 1026
 nppiFilterSobelVertBorder_8u_C1R_Ctx (C function), 1026

<i>function</i>), 1025	<i>function</i>), 1099
nppiFilterSobelVertBorder_8u_C3R_Ctx (C <i>function</i>), 1025	nppiFilterUnsharpBorder_16u_AC4R_Ctx (C <i>function</i>), 1098
nppiFilterSobelVertBorder_8u_C4R_Ctx (C <i>function</i>), 1026	nppiFilterUnsharpBorder_16u_C1R_Ctx (C <i>function</i>), 1098
nppiFilterSobelVertBorder_16s_AC4R_Ctx (C <i>function</i>), 1026	nppiFilterUnsharpBorder_16u_C3R_Ctx (C <i>function</i>), 1098
nppiFilterSobelVertBorder_16s_C1R_Ctx (C <i>function</i>), 1026	nppiFilterUnsharpBorder_16u_C4R_Ctx (C <i>function</i>), 1098
nppiFilterSobelVertBorder_16s_C3R_Ctx (C <i>function</i>), 1026	nppiFilterUnsharpBorder_32f_AC4R_Ctx (C <i>function</i>), 1100
nppiFilterSobelVertBorder_16s_C4R_Ctx (C <i>function</i>), 1026	nppiFilterUnsharpBorder_32f_C1R_Ctx (C <i>function</i>), 1100
nppiFilterSobelVertBorder_32f_AC4R_Ctx (C <i>function</i>), 1027	nppiFilterUnsharpBorder_32f_C3R_Ctx (C <i>function</i>), 1100
nppiFilterSobelVertBorder_32f_C1R_Ctx (C <i>function</i>), 1027	nppiFilterUnsharpBorder_32f_C4R_Ctx (C <i>function</i>), 1100
nppiFilterSobelVertBorder_32f_C3R_Ctx (C <i>function</i>), 1027	nppiFilterUnsharpGetBufferSize_8u_AC4R (C <i>function</i>), 1101
nppiFilterSobelVertBorder_32f_C4R_Ctx (C <i>function</i>), 1027	nppiFilterUnsharpGetBufferSize_8u_C1R (C <i>function</i>), 1101
nppiFilterSobelVertMask_32f_C1R_Ctx (C <i>function</i>), 1019	nppiFilterUnsharpGetBufferSize_8u_C3R (C <i>function</i>), 1101
nppiFilterSobelVertMaskBorder_32f_C1R_Ctx (C <i>function</i>), 1028	nppiFilterUnsharpGetBufferSize_8u_C4R (C <i>function</i>), 1101
nppiFilterSobelVertSecond_8s16s_C1R_Ctx (C <i>function</i>), 1020	nppiFilterUnsharpGetBufferSize_16s_AC4R (C <i>function</i>), 1102
nppiFilterSobelVertSecond_8u16s_C1R_Ctx (C <i>function</i>), 1020	nppiFilterUnsharpGetBufferSize_16s_C1R (C <i>function</i>), 1102
nppiFilterSobelVertSecond_32f_C1R_Ctx (C <i>function</i>), 1020	nppiFilterUnsharpGetBufferSize_16s_C3R (C <i>function</i>), 1102
nppiFilterSobelVertSecondBorder_8s16s_C1R_Ctx (C <i>function</i>), 1030	nppiFilterUnsharpGetBufferSize_16s_C4R (C <i>function</i>), 1102
nppiFilterSobelVertSecondBorder_8u16s_C1R_Ctx (C <i>function</i>), 1030	nppiFilterUnsharpGetBufferSize_16u_AC4R (C <i>function</i>), 1102
nppiFilterSobelVertSecondBorder_32f_C1R_Ctx (C <i>function</i>), 1030	nppiFilterUnsharpGetBufferSize_16u_C1R (C <i>function</i>), 1101
nppiFilterThresholdAdaptiveBoxBorder_8u_C1R_Ctx (C <i>function</i>), 972	nppiFilterUnsharpGetBufferSize_16u_C3R (C <i>function</i>), 1101
nppiFilterUnsharpBorder_8u_AC4R_Ctx (C <i>function</i>), 1098	nppiFilterUnsharpGetBufferSize_16u_C4R (C <i>function</i>), 1101
nppiFilterUnsharpBorder_8u_C1R_Ctx (C <i>function</i>), 1097	nppiFilterUnsharpGetBufferSize_32f_AC4R (C <i>function</i>), 1102
nppiFilterUnsharpBorder_8u_C3R_Ctx (C <i>function</i>), 1097	nppiFilterUnsharpGetBufferSize_32f_C1R (C <i>function</i>), 1102
nppiFilterUnsharpBorder_8u_C4R_Ctx (C <i>function</i>), 1097	nppiFilterUnsharpGetBufferSize_32f_C3R (C <i>function</i>), 1102
nppiFilterUnsharpBorder_16s_AC4R_Ctx (C <i>function</i>), 1099	nppiFilterUnsharpGetBufferSize_32f_C4R (C <i>function</i>), 1102
nppiFilterUnsharpBorder_16s_C1R_Ctx (C <i>function</i>), 1099	nppiFilterWienerBorder_8u_AC4R_Ctx (C <i>function</i>), 1105
nppiFilterUnsharpBorder_16s_C3R_Ctx (C <i>function</i>), 1099	nppiFilterWienerBorder_8u_C1R_Ctx (C <i>function</i>), 1104
nppiFilterUnsharpBorder_16s_C4R_Ctx (C	nppiFilterWienerBorder_8u_C3R_Ctx (C

<i>function</i>), 1104		<i>function</i>), 1200	
nppiFilterWienerBorder_8u_C4R_Ctx (C <i>function</i>), 1104		nppiFloodFillGradient_32u_C3IR_Ctx (C <i>function</i>), 1200	
nppiFilterWienerBorder_16s_AC4R_Ctx (C <i>function</i>), 1106		nppiFloodFillGradientBoundary_8u_C1IR_Ctx (C <i>function</i>), 1202	
nppiFilterWienerBorder_16s_C1R_Ctx (C <i>function</i>), 1105		nppiFloodFillGradientBoundary_8u_C3IR_Ctx (C <i>function</i>), 1202	
nppiFilterWienerBorder_16s_C3R_Ctx (C <i>function</i>), 1105		nppiFloodFillGradientBoundary_16u_C1IR_Ctx (C <i>function</i>), 1203	
nppiFilterWienerBorder_16s_C4R_Ctx (C <i>function</i>), 1105		nppiFloodFillGradientBoundary_16u_C3IR_Ctx (C <i>function</i>), 1204	
nppiFilterWienerBorder_32f_AC4R_Ctx (C <i>function</i>), 1106		nppiFloodFillGradientBoundary_32u_C1IR_Ctx (C <i>function</i>), 1205	
nppiFilterWienerBorder_32f_C1R_Ctx (C <i>function</i>), 1106		nppiFloodFillGradientBoundary_32u_C3IR_Ctx (C <i>function</i>), 1205	
nppiFilterWienerBorder_32f_C3R_Ctx (C <i>function</i>), 1106		nppiFloodFillRange_8u_C1IR_Ctx (C <i>function</i>), 1188	
nppiFilterWienerBorder_32f_C4R_Ctx (C <i>function</i>), 1106		nppiFloodFillRange_8u_C3IR_Ctx (C <i>function</i>), 1189	
nppiFloodFill_8u_C1IR_Ctx (C <i>function</i>), 1180		nppiFloodFillRange_16u_C1IR_Ctx (C <i>function</i>), 1189	
nppiFloodFill_8u_C3IR_Ctx (C <i>function</i>), 1181		nppiFloodFillRange_16u_C3IR_Ctx (C <i>function</i>), 1190	
nppiFloodFill_16u_C1IR_Ctx (C <i>function</i>), 1181		nppiFloodFillRange_32u_C1IR_Ctx (C <i>function</i>), 1191	
nppiFloodFill_16u_C3IR_Ctx (C <i>function</i>), 1182		nppiFloodFillRange_32u_C3IR_Ctx (C <i>function</i>), 1191	
nppiFloodFill_32u_C1IR_Ctx (C <i>function</i>), 1182		nppiFloodFillRangeBoundary_8u_C1IR_Ctx (C <i>function</i>), 1192	
nppiFloodFill_32u_C3IR_Ctx (C <i>function</i>), 1183		nppiFloodFillRangeBoundary_8u_C3IR_Ctx (C <i>function</i>), 1193	
nppiFloodFillBoundary_8u_C1IR_Ctx (C <i>function</i>), 1184		nppiFloodFillRangeBoundary_16u_C1IR_Ctx (C <i>function</i>), 1194	
nppiFloodFillBoundary_8u_C3IR_Ctx (C <i>function</i>), 1185		nppiFloodFillRangeBoundary_16u_C3IR_Ctx (C <i>function</i>), 1194	
nppiFloodFillBoundary_16u_C1IR_Ctx (C <i>function</i>), 1185		nppiFloodFillRangeBoundary_32u_C1IR_Ctx (C <i>function</i>), 1195	
nppiFloodFillBoundary_16u_C3IR_Ctx (C <i>function</i>), 1186		nppiFloodFillRangeBoundary_32u_C3IR_Ctx (C <i>function</i>), 1196	
nppiFloodFillBoundary_32u_C1IR_Ctx (C <i>function</i>), 1186		nppiFree (C <i>function</i>), 1720	
nppiFloodFillBoundary_32u_C3IR_Ctx (C <i>function</i>), 1187		nppiFullNormLevelGetBufferHostSize_8s32f_AC4R_Ctx (C <i>function</i>), 1538	
nppiFloodFillGetBufferSize (C <i>function</i>), 1180		nppiFullNormLevelGetBufferHostSize_8s32f_C1R_Ctx (C <i>function</i>), 1538	
nppiFloodFillGradient_8u_C1IR_Ctx (C <i>function</i>), 1197		nppiFullNormLevelGetBufferHostSize_8s32f_C3R_Ctx (C <i>function</i>), 1538	
nppiFloodFillGradient_8u_C3IR_Ctx (C <i>function</i>), 1198		nppiFullNormLevelGetBufferHostSize_8s32f_C4R_Ctx (C <i>function</i>), 1538	
nppiFloodFillGradient_16u_C1IR_Ctx (C <i>function</i>), 1198		nppiFullNormLevelGetBufferHostSize_8u32f_AC4R_Ctx (C <i>function</i>), 1537	
nppiFloodFillGradient_16u_C3IR_Ctx (C <i>function</i>), 1199		nppiFullNormLevelGetBufferHostSize_8u32f_C1R_Ctx (C <i>function</i>), 1537	
nppiFloodFillGradient_32u_C1IR_Ctx (C		nppiFullNormLevelGetBufferHostSize_8u32f_C3R_Ctx (C <i>function</i>), 1537	

nppiFullNormLevelGetBufferSize_8u32f_C4R_Ctx (C function), 1065
 (C function), 1537
 nppiFullNormLevelGetBufferSize_8u_AC4RSfs_Ctx (C function), 1068
 (C function), 1536
 nppiFullNormLevelGetBufferSize_8u_C1R6s_Ctx (C function), 1292
 (C function), 1535
 nppiFullNormLevelGetBufferSize_8u_C3RSfs_Ctx (C function), 1291
 (C function), 1536
 nppiFullNormLevelGetBufferSize_8u_C4RSfs_Ctx (C function), 1226
 (C function), 1536
 nppiFullNormLevelGetBufferSize_16u32f_Ac4R1R_Ctx (C function), 1259
 (C function), 1539
 nppiFullNormLevelGetBufferSize_16u32f_Ac4R1R_Ctx (C function), 1259
 (C function), 1538
 nppiFullNormLevelGetBufferSize_16u32f_Ac4R1R_Ctx (C function), 544
 (C function), 1538
 nppiFullNormLevelGetBufferSize_16u32f_Ac4R1R_Ctx (C function), 545
 (C function), 1538
 nppiFullNormLevelGetBufferSize_16u32f_Ac4R1R_Ctx (C function), 544
 (C function), 1536
 nppiFullNormLevelGetBufferSize_32f_Ac4R1R_Ctx (C function), 545
 (C function), 1536
 nppiFullNormLevelGetBufferSize_32f_C1R16s_Ctx (C function), 1109
 (C function), 1536
 nppiFullNormLevelGetBufferSize_32f_C3C1R_Ctx (C function), 1109
 (C function), 1536
 nppiFullNormLevelGetBufferSize_32f_C3C1R_Ctx (C function), 1109
 (C function), 1536
 nppiFullNormLevelGetBufferSize_64f_Ac4R1R_Ctx (C function), 1110
 (C function), 1537
 nppiFullNormLevelGetBufferSize_64f_C1R16s_Ctx (C function), 1110
 (C function), 1537
 nppiFullNormLevelGetBufferSize_64f_C3C1R_Ctx (C function), 1111
 (C function), 1537
 nppiFullNormLevelGetBufferSize_64f_C3C1R_Ctx (C function), 1111
 (C function), 1537
 nppiFusedAbsDiff_Threshold_GTVal_Ctx (C function), 1693
 nppiFusedAbsDiff_Threshold_GTVal_I_Ctx (C function), 1693
 nppiGammaFwd_8u_AC4IR_Ctx (C function), 551
 nppiGammaFwd_8u_AC4R_Ctx (C function), 551
 nppiGammaFwd_8u_C3IR_Ctx (C function), 550
 nppiGammaFwd_8u_C3R_Ctx (C function), 550
 nppiGammaFwd_8u_IP3R_Ctx (C function), 552
 nppiGammaFwd_8u_P3R_Ctx (C function), 551
 nppiGammaInv_8u_AC4IR_Ctx (C function), 553
 nppiGammaInv_8u_AC4R_Ctx (C function), 552
 nppiGammaInv_8u_C3IR_Ctx (C function), 552
 nppiGammaInv_8u_C3R_Ctx (C function), 552
 nppiGammaInv_8u_IP3R_Ctx (C function), 553
 nppiGammaInv_8u_P3R_Ctx (C function), 553
 nppiGetAffineBound (C function), 1267
 nppiGetAffineQuad (C function), 1266
 nppiGetAffineTransform (C function), 1266
 nppiGetFilterGaussPyramidLayerDownBorderDstROI (C function), 1120
 nppiGetFilterGaussPyramidLayerUpBorderDstROI (C function), 1065
 nppiGetPerspectiveBound (C function), 1292
 nppiGetPerspectiveQuad (C function), 1292
 nppiGetPerspectiveTransform (C function), 1291
 nppiGetResizeRect (C function), 1226
 nppiGetResizeTiledSourceOffset (C function), 1235
 nppiGetRotateBound (C function), 1259
 nppiGetRotateQuad (C function), 1259
 nppiGradientColorToGray_8u_C3C1R_Ctx (C function), 544
 nppiGradientColorToGray_16s_C3C1R_Ctx (C function), 545
 nppiGradientColorToGray_16u_C3C1R_Ctx (C function), 544
 nppiGradientColorToGray_32f_C3C1R_Ctx (C function), 545
 nppiGradientVectorPrewittBorder_8u16s_C1R_Ctx (C function), 1109
 nppiGradientVectorPrewittBorder_8u16s_C3C1R_Ctx (C function), 1109
 nppiGradientVectorPrewittBorder_16s32f_C1R_Ctx (C function), 1109
 nppiGradientVectorPrewittBorder_16s32f_C3C1R_Ctx (C function), 1110
 nppiGradientVectorPrewittBorder_16u32f_C1R_Ctx (C function), 1110
 nppiGradientVectorPrewittBorder_16u32f_C3C1R_Ctx (C function), 1111
 nppiGradientVectorPrewittBorder_32f_C1R_Ctx (C function), 1111
 nppiGradientVectorPrewittBorder_32f_C3C1R_Ctx (C function), 1112
 nppiGradientVectorScharrBorder_8u16s_C1R_Ctx (C function), 1114
 nppiGradientVectorScharrBorder_8u16s_C3C1R_Ctx (C function), 1115
 nppiGradientVectorScharrBorder_16s32f_C1R_Ctx (C function), 1115
 nppiGradientVectorScharrBorder_16s32f_C3C1R_Ctx (C function), 1116
 nppiGradientVectorScharrBorder_16u32f_C1R_Ctx (C function), 1116
 nppiGradientVectorScharrBorder_16u32f_C3C1R_Ctx (C function), 1116
 nppiGradientVectorScharrBorder_32f_C1R_Ctx (C function), 1117
 nppiGradientVectorScharrBorder_32f_C3C1R_Ctx (C function), 1117
 nppiGradientVectorSobelBorder_8u16s_C1R_Ctx (C function), 1120

nppiGradientVectorSobelBorder_8u16s_C3C1R_Ctx (C function), 1120
 nppiGradientVectorSobelBorder_16s32f_C1R_Ctx (C function), 1121
 nppiGradientVectorSobelBorder_16s32f_C3C1R_Ctx (C function), 1121
 nppiGradientVectorSobelBorder_16u32f_C1R_Ctx (C function), 1122
 nppiGradientVectorSobelBorder_16u32f_C3C1R_Ctx (C function), 1122
 nppiGradientVectorSobelBorder_32f_C1R_Ctx (C function), 1122
 nppiGradientVectorSobelBorder_32f_C3C1R_Ctx (C function), 1123
 nppiGrayDilateBorder_8u_C1R_Ctx (C function), 1323
 nppiGrayDilateBorder_32f_C1R_Ctx (C function), 1323
 nppiGrayErodeBorder_8u_C1R_Ctx (C function), 1334
 nppiGrayErodeBorder_32f_C1R_Ctx (C function), 1334
 NppiHaarBuffer (C struct), 40
 NppiHaarBuffer.haarBuffer (C var), 40
 NppiHaarBuffer.haarBufferSize (C var), 40
 NppiHaarClassifier_32f (C struct), 40
 NppiHaarClassifier_32f.classifiers (C var), 40
 NppiHaarClassifier_32f.classifierSize (C var), 40
 NppiHaarClassifier_32f.classifierStep (C var), 40
 NppiHaarClassifier_32f.counterDevice (C var), 40
 NppiHaarClassifier_32f.numClassifiers (C var), 40
 nppiHistogramEven_8u_AC4R_Ctx (C function), 1490
 nppiHistogramEven_8u_C1R_Ctx (C function), 1490
 nppiHistogramEven_8u_C3R_Ctx (C function), 1490
 nppiHistogramEven_8u_C4R_Ctx (C function), 1490
 nppiHistogramEven_16s_AC4R_Ctx (C function), 1492
 nppiHistogramEven_16s_C1R_Ctx (C function), 1491
 nppiHistogramEven_16s_C3R_Ctx (C function), 1491
 nppiHistogramEven_16s_C4R_Ctx (C function), 1491
 nppiHistogramEven_16u_AC4R_Ctx (C function), 1491
 nppiHistogramEven_16u_C1R_Ctx (C function), 1490
 nppiHistogramEven_16u_C3R_Ctx (C function), 1491
 nppiHistogramEven_16u_C4R_Ctx (C function), 1491
 nppiHistogramEven_16u_C3R_Ctx (C function), 1491
 nppiHistogramEven_16u_C4R_Ctx (C function), 1492
 nppiHistogramEven_16u_C3R_Ctx (C function), 1492
 nppiHistogramEven_16u_C4R_Ctx (C function), 1492
 nppiHistogramEven_16s_AC4R_Ctx (C function), 1494
 nppiHistogramEven_16s_C1R_Ctx (C function), 1493
 nppiHistogramEven_16s_C3R_Ctx (C function), 1493
 nppiHistogramEven_16s_C4R_Ctx (C function), 1494
 nppiHistogramEven_16u_AC4R_Ctx (C function), 1493
 nppiHistogramEven_16u_C1R_Ctx (C function), 1493
 nppiHistogramEven_16u_C3R_Ctx (C function), 1493
 nppiHistogramEven_16u_C4R_Ctx (C function), 1493
 nppiHistogramOfGradientsBorder_8u32f_C1R_Ctx (C function), 1176
 nppiHistogramOfGradientsBorder_8u32f_C3R_Ctx (C function), 1177
 nppiHistogramOfGradientsBorder_16s32f_C1R_Ctx (C function), 1178
 nppiHistogramOfGradientsBorder_16s32f_C3R_Ctx (C function), 1178
 nppiHistogramOfGradientsBorder_16u32f_C1R_Ctx (C function), 1177
 nppiHistogramOfGradientsBorder_16u32f_C3R_Ctx (C function), 1177
 nppiHistogramOfGradientsBorder_32f_C1R_Ctx (C function), 1179
 nppiHistogramOfGradientsBorder_32f_C3R_Ctx (C function), 1179
 nppiHistogramOfGradientsBorderGetBufferSize (C function), 1175
 nppiHistogramOfGradientsBorderGetDescriptorsSize (C function), 1176
 nppiHistogramRange_8u_AC4R_Ctx (C function), 1495
 nppiHistogramRange_8u_C1R_Ctx (C function), 1495

nppiHistogramRange_8u_C3R_Ctx (C function), 1495	nppiHistogramRangeGetBufferSize_32f_C1R_Ctx (C function), 1499
nppiHistogramRange_8u_C4R_Ctx (C function), 1495	nppiHistogramRangeGetBufferSize_32f_C3R_Ctx (C function), 1499
nppiHistogramRange_16s_AC4R_Ctx (C function), 1496	nppiHistogramRangeGetBufferSize_32f_C4R_Ctx (C function), 1500
nppiHistogramRange_16s_C1R_Ctx (C function), 1496	nppiHLSToBGR_8u_AC4P4R_Ctx (C function), 787
nppiHistogramRange_16s_C3R_Ctx (C function), 1496	nppiHLSToBGR_8u_AP4C4R_Ctx (C function), 789
nppiHistogramRange_16s_C4R_Ctx (C function), 1496	nppiHLSToBGR_8u_AP4R_Ctx (C function), 788
nppiHistogramRange_16u_AC4R_Ctx (C function), 1496	nppiHLSToBGR_8u_C3P3R_Ctx (C function), 787
nppiHistogramRange_16u_C1R_Ctx (C function), 1495	nppiHLSToBGR_8u_P3C3R_Ctx (C function), 788
nppiHistogramRange_16u_C3R_Ctx (C function), 1496	nppiHLSToBGR_8u_P3R_Ctx (C function), 788
nppiHistogramRange_16u_C4R_Ctx (C function), 1496	nppiHLSToRGB_8u_AC4R_Ctx (C function), 784
nppiHistogramRange_32f_AC4R_Ctx (C function), 1497	nppiHLSToRGB_8u_C3R_Ctx (C function), 783
nppiHistogramRange_32f_C1R_Ctx (C function), 1497	NppiHOGConfig (C struct), 39
nppiHistogramRange_32f_C3R_Ctx (C function), 1497	NppiHOGConfig.cellSize (C var), 39
nppiHistogramRange_32f_C4R_Ctx (C function), 1497	NppiHOGConfig.detectionWindowSize (C var), 39
nppiHistogramRangeGetBufferSize_8u_AC4R_Ctx (C function), 1498	NppiHOGConfig.histogramBlockSize (C var), 39
nppiHistogramRangeGetBufferSize_8u_C1R_Ctx (C function), 1497	NppiHOGConfig.nHistogramBins (C var), 39
nppiHistogramRangeGetBufferSize_8u_C3R_Ctx (C function), 1498	nppiHSVToRGB_8u_AC4R_Ctx (C function), 792
nppiHistogramRangeGetBufferSize_8u_C4R_Ctx (C function), 1498	nppiHSVToRGB_8u_C3R_Ctx (C function), 792
nppiHistogramRangeGetBufferSize_16s_AC4R_Ctx (C function), 1499	NppiHuffmanTableType (C enum), 41
nppiHistogramRangeGetBufferSize_16s_C1R_Ctx (C function), 1499	NppiHuffmanTableType.nppiACTable (C enumerator), 41
nppiHistogramRangeGetBufferSize_16s_C3R_Ctx (C function), 1499	NppiHuffmanTableType.nppiDCTable (C enumerator), 41
nppiHistogramRangeGetBufferSize_16s_C4R_Ctx (C function), 1499	nppiI444_8u_ToUYVP_10u_P3C3R_Ctx (C function), 532
nppiHistogramRangeGetBufferSize_16u_AC4R_Ctx (C function), 1498	NppiImageDescriptor (C struct), 41
nppiHistogramRangeGetBufferSize_16u_C1R_Ctx (C function), 1498	NppiImageDescriptor.nStep (C var), 42
nppiHistogramRangeGetBufferSize_16u_C3R_Ctx (C function), 1498	NppiImageDescriptor.oSize (C var), 42
nppiHistogramRangeGetBufferSize_16u_C4R_Ctx (C function), 1498	NppiImageDescriptor.pData (C var), 42
nppiHistogramRangeGetBufferSize_32f_AC4R_Ctx (C function), 1500	nppiIntegral_8u32f_C1R_Ctx (C function), 1484
	nppiIntegral_8u32s_C1R_Ctx (C function), 1484
	NppiInterpolationMode (C enum), 21
	NppiInterpolationMode.NPPI_INTER_CUBIC (C enumerator), 21
	NppiInterpolationMode.NPPI_INTER_CUBIC2P_B05C03 (C enumerator), 21
	NppiInterpolationMode.NPPI_INTER_CUBIC2P_BSPLINE (C enumerator), 21
	NppiInterpolationMode.NPPI_INTER_CUBIC2P_CATMULLROM (C enumerator), 21
	NppiInterpolationMode.NPPI_INTER_LANCZOS (C enumerator), 21
	NppiInterpolationMode.NPPI_INTER_LANCZOS3_ADVANCED (C enumerator), 21
	NppiInterpolationMode.NPPI_INTER_LINEAR (C enumerator), 21

(C enumerator), 21

NppiInterpolationMode.NPPI_INTER_NN (C enumerator), 21

NppiInterpolationMode.NPPI_INTER_SUPER (C enumerator), 21

NppiInterpolationMode.NPPI_INTER_UNDEFINED (C enumerator), 21

NppiInterpolationMode.NPPI_SMOOTH_EDGE (C enumerator), 21

nppiLabelMarkersUF_8u32u_C1R_Ctx (C function), 1208

nppiLabelMarkersUF_16u32u_C1R_Ctx (C function), 1208

nppiLabelMarkersUF_32u_C1R_Ctx (C function), 1208

nppiLabelMarkersUFBatch_8u32u_C1R_Advanced (C function), 1212

nppiLabelMarkersUFBatch_8u32u_C1R_Ctx (C function), 1210

nppiLabelMarkersUFBatch_16u32u_C1R_Advanced (C function), 1212

nppiLabelMarkersUFBatch_16u32u_C1R_Ctx (C function), 1210

nppiLabelMarkersUFBatch_32u_C1R_Advanced (C function), 1213

nppiLabelMarkersUFBatch_32u_C1R_Ctx (C function), 1211

nppiLabelMarkersUFGetBufferSize_32u_C1R (C function), 1207

nppiLabToBGR_8u_C3R_Ctx (C function), 775

nppiLn_8u_C1RSfs_Ctx (C function), 400

nppiLn_8u_C1RSfs_Ctx (C function), 400

nppiLn_8u_C3RSfs_Ctx (C function), 401

nppiLn_8u_C3RSfs_Ctx (C function), 400

nppiLn_16f_C1IR_Ctx (C function), 404

nppiLn_16f_C1R_Ctx (C function), 404

nppiLn_16f_C3IR_Ctx (C function), 405

nppiLn_16f_C3R_Ctx (C function), 405

nppiLn_16s_C1RSfs_Ctx (C function), 403

nppiLn_16s_C1RSfs_Ctx (C function), 403

nppiLn_16s_C3RSfs_Ctx (C function), 404

nppiLn_16s_C3RSfs_Ctx (C function), 403

nppiLn_16u_C1RSfs_Ctx (C function), 402

nppiLn_16u_C1RSfs_Ctx (C function), 401

nppiLn_16u_C3RSfs_Ctx (C function), 402

nppiLn_16u_C3RSfs_Ctx (C function), 402

nppiLn_32f_C1IR_Ctx (C function), 406

nppiLn_32f_C1R_Ctx (C function), 405

nppiLn_32f_C3IR_Ctx (C function), 406

nppiLn_32f_C3R_Ctx (C function), 406

nppiLShiftC_8u_AC4IR_Ctx (C function), 454

nppiLShiftC_8u_AC4R_Ctx (C function), 454

nppiLShiftC_8u_C1IR_Ctx (C function), 453

nppiLShiftC_8u_C1R_Ctx (C function), 452

nppiLShiftC_8u_C3IR_Ctx (C function), 453

nppiLShiftC_8u_C3R_Ctx (C function), 453

nppiLShiftC_8u_C4IR_Ctx (C function), 455

nppiLShiftC_8u_C4R_Ctx (C function), 455

nppiLShiftC_16u_AC4IR_Ctx (C function), 457

nppiLShiftC_16u_AC4R_Ctx (C function), 457

nppiLShiftC_16u_C1IR_Ctx (C function), 456

nppiLShiftC_16u_C1R_Ctx (C function), 455

nppiLShiftC_16u_C3IR_Ctx (C function), 456

nppiLShiftC_16u_C3R_Ctx (C function), 456

nppiLShiftC_16u_C4IR_Ctx (C function), 458

nppiLShiftC_16u_C4R_Ctx (C function), 457

nppiLShiftC_32s_AC4IR_Ctx (C function), 460

nppiLShiftC_32s_AC4R_Ctx (C function), 459

nppiLShiftC_32s_C1IR_Ctx (C function), 458

nppiLShiftC_32s_C1R_Ctx (C function), 458

nppiLShiftC_32s_C3IR_Ctx (C function), 459

nppiLShiftC_32s_C3R_Ctx (C function), 459

nppiLShiftC_32s_C4IR_Ctx (C function), 460

nppiLShiftC_32s_C4R_Ctx (C function), 460

nppiLUT_8u_AC4IR_Ctx (C function), 606

nppiLUT_8u_AC4R_Ctx (C function), 605

nppiLUT_8u_C1IR_Ctx (C function), 602

nppiLUT_8u_C1R_Ctx (C function), 602

nppiLUT_8u_C3IR_Ctx (C function), 604

nppiLUT_8u_C3R_Ctx (C function), 603

nppiLUT_8u_C4IR_Ctx (C function), 605

nppiLUT_8u_C4R_Ctx (C function), 604

nppiLUT_16s_AC4IR_Ctx (C function), 614

nppiLUT_16s_AC4R_Ctx (C function), 614

nppiLUT_16s_C1IR_Ctx (C function), 611

nppiLUT_16s_C1R_Ctx (C function), 611

nppiLUT_16s_C3IR_Ctx (C function), 612

nppiLUT_16s_C3R_Ctx (C function), 612

nppiLUT_16s_C4IR_Ctx (C function), 613

nppiLUT_16s_C4R_Ctx (C function), 613

nppiLUT_16u_AC4IR_Ctx (C function), 610

nppiLUT_16u_AC4R_Ctx (C function), 609

nppiLUT_16u_C1IR_Ctx (C function), 607

nppiLUT_16u_C1R_Ctx (C function), 606

nppiLUT_16u_C3IR_Ctx (C function), 608

nppiLUT_16u_C3R_Ctx (C function), 607

nppiLUT_16u_C4IR_Ctx (C function), 609

nppiLUT_16u_C4R_Ctx (C function), 608

nppiLUT_32f_AC4IR_Ctx (C function), 619

nppiLUT_32f_AC4R_Ctx (C function), 618

nppiLUT_32f_C1IR_Ctx (C function), 615

nppiLUT_32f_C1R_Ctx (C function), 615

nppiLUT_32f_C3IR_Ctx (C function), 617

nppiLUT_32f_C3R_Ctx (C function), 616

nppiLUT_32f_C4IR_Ctx (C function), 618

nppiLUT_32f_C4R_Ctx (C function), 617

nppiLUT_Cubic_8u_AC4IR_Ctx (C function), 642

nppiLUT_Cubic_8u_AC4R_Ctx (C function), 642
 nppiLUT_Cubic_8u_C1IR_Ctx (C function), 639
 nppiLUT_Cubic_8u_C1R_Ctx (C function), 639
 nppiLUT_Cubic_8u_C3IR_Ctx (C function), 640
 nppiLUT_Cubic_8u_C3R_Ctx (C function), 640
 nppiLUT_Cubic_8u_C4IR_Ctx (C function), 641
 nppiLUT_Cubic_8u_C4R_Ctx (C function), 641
 nppiLUT_Cubic_16s_AC4IR_Ctx (C function), 651
 nppiLUT_Cubic_16s_AC4R_Ctx (C function), 651
 nppiLUT_Cubic_16s_C1IR_Ctx (C function), 648
 nppiLUT_Cubic_16s_C1R_Ctx (C function), 647
 nppiLUT_Cubic_16s_C3IR_Ctx (C function), 649
 nppiLUT_Cubic_16s_C3R_Ctx (C function), 648
 nppiLUT_Cubic_16s_C4IR_Ctx (C function), 650
 nppiLUT_Cubic_16s_C4R_Ctx (C function), 649
 nppiLUT_Cubic_16u_AC4IR_Ctx (C function), 647
 nppiLUT_Cubic_16u_AC4R_Ctx (C function), 646
 nppiLUT_Cubic_16u_C1IR_Ctx (C function), 644
 nppiLUT_Cubic_16u_C1R_Ctx (C function), 643
 nppiLUT_Cubic_16u_C3IR_Ctx (C function), 645
 nppiLUT_Cubic_16u_C3R_Ctx (C function), 644
 nppiLUT_Cubic_16u_C4IR_Ctx (C function), 646
 nppiLUT_Cubic_16u_C4R_Ctx (C function), 645
 nppiLUT_Cubic_32f_AC4IR_Ctx (C function), 656
 nppiLUT_Cubic_32f_AC4R_Ctx (C function), 655
 nppiLUT_Cubic_32f_C1IR_Ctx (C function), 652
 nppiLUT_Cubic_32f_C1R_Ctx (C function), 652
 nppiLUT_Cubic_32f_C3IR_Ctx (C function), 653
 nppiLUT_Cubic_32f_C3R_Ctx (C function), 653
 nppiLUT_Cubic_32f_C4IR_Ctx (C function), 654
 nppiLUT_Cubic_32f_C4R_Ctx (C function), 654
 nppiLUT_Linear_8u_AC4IR_Ctx (C function), 624
 nppiLUT_Linear_8u_AC4R_Ctx (C function), 624
 nppiLUT_Linear_8u_C1IR_Ctx (C function), 620
 nppiLUT_Linear_8u_C1R_Ctx (C function), 620
 nppiLUT_Linear_8u_C3IR_Ctx (C function), 622
 nppiLUT_Linear_8u_C3R_Ctx (C function), 621
 nppiLUT_Linear_8u_C4IR_Ctx (C function), 623
 nppiLUT_Linear_8u_C4R_Ctx (C function), 622
 nppiLUT_Linear_16s_AC4IR_Ctx (C function), 633
 nppiLUT_Linear_16s_AC4R_Ctx (C function), 633
 nppiLUT_Linear_16s_C1IR_Ctx (C function), 630
 nppiLUT_Linear_16s_C1R_Ctx (C function), 629
 nppiLUT_Linear_16s_C3IR_Ctx (C function), 631
 nppiLUT_Linear_16s_C3R_Ctx (C function), 630
 nppiLUT_Linear_16s_C4IR_Ctx (C function), 632
 nppiLUT_Linear_16s_C4R_Ctx (C function), 631
 nppiLUT_Linear_16u_AC4IR_Ctx (C function), 629
 nppiLUT_Linear_16u_AC4R_Ctx (C function), 628
 nppiLUT_Linear_16u_C1IR_Ctx (C function), 626
 nppiLUT_Linear_16u_C1R_Ctx (C function), 625
 nppiLUT_Linear_16u_C3IR_Ctx (C function), 627
 nppiLUT_Linear_16u_C3R_Ctx (C function), 626
 nppiLUT_Linear_16u_C4IR_Ctx (C function), 628
 nppiLUT_Linear_16u_C4R_Ctx (C function), 627
 nppiLUT_Linear_32f_AC4IR_Ctx (C function), 638
 nppiLUT_Linear_32f_AC4R_Ctx (C function), 637
 nppiLUT_Linear_32f_C1IR_Ctx (C function), 634
 nppiLUT_Linear_32f_C1R_Ctx (C function), 634
 nppiLUT_Linear_32f_C3IR_Ctx (C function), 635
 nppiLUT_Linear_32f_C3R_Ctx (C function), 635
 nppiLUT_Linear_32f_C4IR_Ctx (C function), 637
 nppiLUT_Linear_32f_C4R_Ctx (C function), 636
 nppiLUT_Trilinear_8u_AC4IR_Ctx (C func-

tion), 658
 nppiLUT_Trilinear_8u_AC4R_Ctx (C function), 657
 nppiLUT_Trilinear_8u_C4R_Ctx (C function), 656
 nppiLUTPalette_8u24u_C1R_Ctx (C function), 659
 nppiLUTPalette_8u32u_C1R_Ctx (C function), 660
 nppiLUTPalette_8u_AC4R_Ctx (C function), 661
 nppiLUTPalette_8u_C1R_Ctx (C function), 659
 nppiLUTPalette_8u_C3R_Ctx (C function), 660
 nppiLUTPalette_8u_C4R_Ctx (C function), 661
 nppiLUTPalette_16u8u_C1R_Ctx (C function), 662
 nppiLUTPalette_16u24u_C1R_Ctx (C function), 663
 nppiLUTPalette_16u32u_C1R_Ctx (C function), 663
 nppiLUTPalette_16u_AC4R_Ctx (C function), 665
 nppiLUTPalette_16u_C1R_Ctx (C function), 662
 nppiLUTPalette_16u_C3R_Ctx (C function), 664
 nppiLUTPalette_16u_C4R_Ctx (C function), 665
 nppiLUTPaletteSwap_8u_C3A0C4R_Ctx (C function), 666
 nppiLUTPaletteSwap_16u_C3A0C4R_Ctx (C function), 666
 nppiLUVToRGB_8u_AC4R_Ctx (C function), 772
 nppiLUVToRGB_8u_C3R_Ctx (C function), 772
 nppiMagnitude_32fc32f_C1R_Ctx (C function), 1316
 nppiMagnitudeSqr_32fc32f_C1R_Ctx (C function), 1316
 nppiMalloc_8u_C1 (C function), 1713
 nppiMalloc_8u_C2 (C function), 1713
 nppiMalloc_8u_C3 (C function), 1713
 nppiMalloc_8u_C4 (C function), 1713
 nppiMalloc_16s_C1 (C function), 1714
 nppiMalloc_16s_C2 (C function), 1715
 nppiMalloc_16s_C4 (C function), 1715
 nppiMalloc_16sc_C1 (C function), 1715
 nppiMalloc_16sc_C2 (C function), 1715
 nppiMalloc_16sc_C3 (C function), 1716
 nppiMalloc_16sc_C4 (C function), 1716
 nppiMalloc_16u_C1 (C function), 1713
 nppiMalloc_16u_C2 (C function), 1714
 nppiMalloc_16u_C3 (C function), 1714
 nppiMalloc_16u_C4 (C function), 1714
 nppiMalloc_32f_C1 (C function), 1718
 nppiMalloc_32f_C2 (C function), 1718
 nppiMalloc_32f_C3 (C function), 1718
 nppiMalloc_32f_C4 (C function), 1718
 nppiMalloc_32fc_C1 (C function), 1719
 nppiMalloc_32fc_C2 (C function), 1719
 nppiMalloc_32fc_C3 (C function), 1719
 nppiMalloc_32fc_C4 (C function), 1719
 nppiMalloc_32s_C1 (C function), 1716
 nppiMalloc_32s_C3 (C function), 1716
 nppiMalloc_32s_C4 (C function), 1716
 nppiMalloc_32sc_C1 (C function), 1717
 nppiMalloc_32sc_C2 (C function), 1717
 nppiMalloc_32sc_C3 (C function), 1717
 nppiMalloc_32sc_C4 (C function), 1717
 NppiMaskSize (C enum), 22
 NppiMaskSize.NPP_MASK_SIZE_1_X_3 (C enumerator), 22
 NppiMaskSize.NPP_MASK_SIZE_1_X_5 (C enumerator), 22
 NppiMaskSize.NPP_MASK_SIZE_3_X_1 (C enumerator), 22
 NppiMaskSize.NPP_MASK_SIZE_3_X_3 (C enumerator), 22
 NppiMaskSize.NPP_MASK_SIZE_5_X_1 (C enumerator), 22
 NppiMaskSize.NPP_MASK_SIZE_5_X_5 (C enumerator), 22
 NppiMaskSize.NPP_MASK_SIZE_7_X_7 (C enumerator), 22
 NppiMaskSize.NPP_MASK_SIZE_9_X_9 (C enumerator), 22
 NppiMaskSize.NPP_MASK_SIZE_11_X_11 (C enumerator), 22
 NppiMaskSize.NPP_MASK_SIZE_13_X_13 (C enumerator), 23
 NppiMaskSize.NPP_MASK_SIZE_15_X_15 (C enumerator), 23
 nppiMax_8u_AC4R_Ctx (C function), 1374
 nppiMax_8u_C1R_Ctx (C function), 1373
 nppiMax_8u_C3R_Ctx (C function), 1373
 nppiMax_8u_C4R_Ctx (C function), 1373
 nppiMax_16s_AC4R_Ctx (C function), 1374
 nppiMax_16s_C1R_Ctx (C function), 1373
 nppiMax_16s_C3R_Ctx (C function), 1373
 nppiMax_16s_C4R_Ctx (C function), 1374
 nppiMax_16u_AC4R_Ctx (C function), 1374
 nppiMax_16u_C1R_Ctx (C function), 1373
 nppiMax_16u_C3R_Ctx (C function), 1373
 nppiMax_16u_C4R_Ctx (C function), 1374
 nppiMax_32f_AC4R_Ctx (C function), 1374
 nppiMax_32f_C1R_Ctx (C function), 1373
 nppiMax_32f_C3R_Ctx (C function), 1373
 nppiMax_32f_C4R_Ctx (C function), 1374

nppiMaxEvery_8u_AC4IR_Ctx (C function), 1480	(C function), 1376
nppiMaxEvery_8u_C1IR_Ctx (C function), 1478	nppiMaxGetBufferHostSize_32f_C1R_Ctx (C function), 1375
nppiMaxEvery_8u_C3IR_Ctx (C function), 1479	nppiMaxGetBufferHostSize_32f_C3R_Ctx (C function), 1376
nppiMaxEvery_8u_C4IR_Ctx (C function), 1479	nppiMaxGetBufferHostSize_32f_C4R_Ctx (C function), 1376
nppiMaxEvery_16s_AC4IR_Ctx (C function), 1480	nppiMaximumError_8s_C1R_Ctx (C function), 1580
nppiMaxEvery_16s_C1IR_Ctx (C function), 1478	nppiMaximumError_8s_C2R_Ctx (C function), 1581
nppiMaxEvery_16s_C3IR_Ctx (C function), 1479	nppiMaximumError_8s_C3R_Ctx (C function), 1583
nppiMaxEvery_16s_C4IR_Ctx (C function), 1480	nppiMaximumError_8s_C4R_Ctx (C function), 1585
nppiMaxEvery_16u_AC4IR_Ctx (C function), 1480	nppiMaximumError_8u_C1R_Ctx (C function), 1580
nppiMaxEvery_16u_C1IR_Ctx (C function), 1478	nppiMaximumError_8u_C2R_Ctx (C function), 1581
nppiMaxEvery_16u_C3IR_Ctx (C function), 1479	nppiMaximumError_8u_C3R_Ctx (C function), 1583
nppiMaxEvery_16u_C4IR_Ctx (C function), 1479	nppiMaximumError_8u_C4R_Ctx (C function), 1585
nppiMaxEvery_32f_AC4IR_Ctx (C function), 1480	nppiMaximumError_16s_C1R_Ctx (C function), 1580
nppiMaxEvery_32f_C1IR_Ctx (C function), 1479	nppiMaximumError_16s_C2R_Ctx (C function), 1582
nppiMaxEvery_32f_C3IR_Ctx (C function), 1479	nppiMaximumError_16s_C3R_Ctx (C function), 1583
nppiMaxEvery_32f_C4IR_Ctx (C function), 1480	nppiMaximumError_16s_C4R_Ctx (C function), 1585
nppiMaxGetBufferHostSize_8u_AC4R_Ctx (C function), 1376	nppiMaximumError_16sc_C1R_Ctx (C func- tion), 1580
nppiMaxGetBufferHostSize_8u_C1R_Ctx (C function), 1375	nppiMaximumError_16sc_C2R_Ctx (C func- tion), 1582
nppiMaxGetBufferHostSize_8u_C3R_Ctx (C function), 1375	nppiMaximumError_16sc_C3R_Ctx (C func- tion), 1584
nppiMaxGetBufferHostSize_8u_C4R_Ctx (C function), 1376	nppiMaximumError_16sc_C4R_Ctx (C func- tion), 1585
nppiMaxGetBufferHostSize_16s_AC4R_Ctx (C function), 1376	nppiMaximumError_16u_C1R_Ctx (C function), 1580
nppiMaxGetBufferHostSize_16s_C1R_Ctx (C function), 1375	nppiMaximumError_16u_C2R_Ctx (C function), 1582
nppiMaxGetBufferHostSize_16s_C3R_Ctx (C function), 1375	nppiMaximumError_16u_C3R_Ctx (C function), 1583
nppiMaxGetBufferHostSize_16s_C4R_Ctx (C function), 1376	nppiMaximumError_16u_C4R_Ctx (C function), 1585
nppiMaxGetBufferHostSize_16u_AC4R_Ctx (C function), 1376	nppiMaximumError_32f_C1R_Ctx (C function), 1581
nppiMaxGetBufferHostSize_16u_C1R_Ctx (C function), 1375	nppiMaximumError_32f_C2R_Ctx (C function), 1582
nppiMaxGetBufferHostSize_16u_C3R_Ctx (C function), 1375	nppiMaximumError_32f_C3R_Ctx (C function), 1584
nppiMaxGetBufferHostSize_16u_C4R_Ctx (C function), 1376	nppiMaximumError_32f_C4R_Ctx (C function),

1586
 nppiMaximumError_32fc_C1R_Ctx (C function), 1581
 nppiMaximumError_32fc_C2R_Ctx (C function), 1583
 nppiMaximumError_32fc_C3R_Ctx (C function), 1584
 nppiMaximumError_32fc_C4R_Ctx (C function), 1586
 nppiMaximumError_32s_C1R_Ctx (C function), 1580
 nppiMaximumError_32s_C2R_Ctx (C function), 1582
 nppiMaximumError_32s_C3R_Ctx (C function), 1584
 nppiMaximumError_32s_C4R_Ctx (C function), 1586
 nppiMaximumError_32sc_C1R_Ctx (C function), 1581
 nppiMaximumError_32sc_C2R_Ctx (C function), 1582
 nppiMaximumError_32sc_C3R_Ctx (C function), 1584
 nppiMaximumError_32sc_C4R_Ctx (C function), 1586
 nppiMaximumError_32u_C1R_Ctx (C function), 1580
 nppiMaximumError_32u_C2R_Ctx (C function), 1582
 nppiMaximumError_32u_C3R_Ctx (C function), 1584
 nppiMaximumError_32u_C4R_Ctx (C function), 1585
 nppiMaximumError_64f_C1R_Ctx (C function), 1581
 nppiMaximumError_64f_C2R_Ctx (C function), 1583
 nppiMaximumError_64f_C3R_Ctx (C function), 1584
 nppiMaximumError_64f_C4R_Ctx (C function), 1586
 nppiMaximumErrorGetBufferHostSize_8s_C1R_Ctx (C function), 1587
 nppiMaximumErrorGetBufferHostSize_8s_C2R_Ctx (C function), 1589
 nppiMaximumErrorGetBufferHostSize_8s_C3R_Ctx (C function), 1590
 nppiMaximumErrorGetBufferHostSize_8s_C4R_Ctx (C function), 1591
 nppiMaximumErrorGetBufferHostSize_8u_C1R_Ctx (C function), 1587
 nppiMaximumErrorGetBufferHostSize_8u_C2R_Ctx (C function), 1588
 nppiMaximumErrorGetBufferHostSize_8u_C3R_Ctx (C function), 1590
 nppiMaximumErrorGetBufferHostSize_8u_C4R_Ctx (C function), 1590
 nppiMaximumErrorGetBufferHostSize_16s_C1R_Ctx (C function), 1591
 nppiMaximumErrorGetBufferHostSize_16s_C2R_Ctx (C function), 1587
 nppiMaximumErrorGetBufferHostSize_16s_C3R_Ctx (C function), 1589
 nppiMaximumErrorGetBufferHostSize_16s_C4R_Ctx (C function), 1592
 nppiMaximumErrorGetBufferHostSize_16sc_C1R_Ctx (C function), 1587
 nppiMaximumErrorGetBufferHostSize_16sc_C2R_Ctx (C function), 1589
 nppiMaximumErrorGetBufferHostSize_16sc_C3R_Ctx (C function), 1590
 nppiMaximumErrorGetBufferHostSize_16sc_C4R_Ctx (C function), 1592
 nppiMaximumErrorGetBufferHostSize_16u_C1R_Ctx (C function), 1587
 nppiMaximumErrorGetBufferHostSize_16u_C2R_Ctx (C function), 1589
 nppiMaximumErrorGetBufferHostSize_16u_C3R_Ctx (C function), 1590
 nppiMaximumErrorGetBufferHostSize_16u_C4R_Ctx (C function), 1592
 nppiMaximumErrorGetBufferHostSize_32f_C1R_Ctx (C function), 1588
 nppiMaximumErrorGetBufferHostSize_32f_C2R_Ctx (C function), 1589
 nppiMaximumErrorGetBufferHostSize_32f_C3R_Ctx (C function), 1591
 nppiMaximumErrorGetBufferHostSize_32f_C4R_Ctx (C function), 1592
 nppiMaximumErrorGetBufferHostSize_32fc_C1R_Ctx (C function), 1588
 nppiMaximumErrorGetBufferHostSize_32fc_C2R_Ctx (C function), 1590
 nppiMaximumErrorGetBufferHostSize_32fc_C3R_Ctx (C function), 1591
 nppiMaximumErrorGetBufferHostSize_32fc_C4R_Ctx (C function), 1593
 nppiMaximumErrorGetBufferHostSize_32s_C1R_Ctx (C function), 1588
 nppiMaximumErrorGetBufferHostSize_32s_C2R_Ctx (C function), 1589
 nppiMaximumErrorGetBufferHostSize_32s_C3R_Ctx (C function), 1591
 nppiMaximumErrorGetBufferHostSize_32s_C4R_Ctx (C function), 1592
 nppiMaximumErrorGetBufferHostSize_32sc_C1R_Ctx (C function), 1588
 nppiMaximumErrorGetBufferHostSize_32sc_C2R_Ctx (C function), 1588

(C function), 1589
 nppiMaximumErrorGetBufferHostSize_32sc_C1R_Ctx (C
 (C function), 1591
 nppiMaximumErrorGetBufferHostSize_32sc_C3R_Ctx (C
 (C function), 1592
 nppiMaximumErrorGetBufferHostSize_32u_C1R_Ctx (C
 (C function), 1588
 nppiMaximumErrorGetBufferHostSize_32u_C2R_Ctx (C
 (C function), 1589
 nppiMaximumErrorGetBufferHostSize_32u_C3R_Ctx (C
 (C function), 1591
 nppiMaximumErrorGetBufferHostSize_32u_C4R_Ctx (C
 (C function), 1592
 nppiMaximumErrorGetBufferHostSize_64f_C1R_Ctx (C
 (C function), 1588
 nppiMaximumErrorGetBufferHostSize_64f_C2R_Ctx (C
 (C function), 1590
 nppiMaximumErrorGetBufferHostSize_64f_C3R_Ctx (C
 (C function), 1591
 nppiMaximumErrorGetBufferHostSize_64f_C4R_Ctx (C
 (C function), 1593
 nppiMaximumRelativeError_8s_C1R_Ctx (C
 function), 1607
 nppiMaximumRelativeError_8s_C2R_Ctx (C
 function), 1609
 nppiMaximumRelativeError_8s_C3R_Ctx (C
 function), 1611
 nppiMaximumRelativeError_8s_C4R_Ctx (C
 function), 1613
 nppiMaximumRelativeError_8u_C1R_Ctx (C
 function), 1607
 nppiMaximumRelativeError_8u_C2R_Ctx (C
 function), 1608
 nppiMaximumRelativeError_8u_C3R_Ctx (C
 function), 1610
 nppiMaximumRelativeError_8u_C4R_Ctx (C
 function), 1612
 nppiMaximumRelativeError_16s_C1R_Ctx (C
 function), 1607
 nppiMaximumRelativeError_16s_C2R_Ctx (C
 function), 1609
 nppiMaximumRelativeError_16s_C3R_Ctx (C
 function), 1611
 nppiMaximumRelativeError_16s_C4R_Ctx (C
 function), 1613
 nppiMaximumRelativeError_16sc_C1R_Ctx
 (C function), 1607
 nppiMaximumRelativeError_16sc_C2R_Ctx
 (C function), 1609
 nppiMaximumRelativeError_16sc_C3R_Ctx
 (C function), 1611
 nppiMaximumRelativeError_16sc_C4R_Ctx
 (C function), 1613
 nppiMaximumRelativeError_16u_C1R_Ctx (C
 function), 1607
 nppiMaximumRelativeError_16u_C2R_Ctx (C
 function), 1609
 nppiMaximumRelativeError_16u_C3R_Ctx (C
 function), 1611
 nppiMaximumRelativeError_16u_C4R_Ctx (C
 function), 1613
 nppiMaximumRelativeError_32f_C1R_Ctx (C
 function), 1608
 nppiMaximumRelativeError_32f_C2R_Ctx (C
 function), 1610
 nppiMaximumRelativeError_32f_C3R_Ctx (C
 function), 1612
 nppiMaximumRelativeError_32f_C4R_Ctx (C
 function), 1614
 nppiMaximumRelativeError_32fc_C1R_Ctx
 (C function), 1608
 nppiMaximumRelativeError_32fc_C2R_Ctx
 (C function), 1610
 nppiMaximumRelativeError_32fc_C3R_Ctx
 (C function), 1612
 nppiMaximumRelativeError_32fc_C4R_Ctx
 (C function), 1614
 nppiMaximumRelativeError_32s_C1R_Ctx (C
 function), 1608
 nppiMaximumRelativeError_32s_C2R_Ctx (C
 function), 1610
 nppiMaximumRelativeError_32s_C3R_Ctx (C
 function), 1611
 nppiMaximumRelativeError_32s_C4R_Ctx (C
 function), 1613
 nppiMaximumRelativeError_32sc_C1R_Ctx
 (C function), 1608
 nppiMaximumRelativeError_32sc_C2R_Ctx
 (C function), 1610
 nppiMaximumRelativeError_32sc_C3R_Ctx
 (C function), 1612
 nppiMaximumRelativeError_32sc_C4R_Ctx
 (C function), 1614
 nppiMaximumRelativeError_32u_C1R_Ctx (C
 function), 1607
 nppiMaximumRelativeError_32u_C2R_Ctx (C
 function), 1609
 nppiMaximumRelativeError_32u_C3R_Ctx (C
 function), 1611
 nppiMaximumRelativeError_32u_C4R_Ctx (C
 function), 1613
 nppiMaximumRelativeError_64f_C1R_Ctx (C
 function), 1608
 nppiMaximumRelativeError_64f_C2R_Ctx (C
 function), 1610
 nppiMaximumRelativeError_64f_C3R_Ctx (C
 function), 1612
 nppiMaximumRelativeError_64f_C4R_Ctx (C

function), 1614
 (C function), 1615
 nppiMaximumRelativeErrorGetBufferHostSize_32fc_C4R_Ctx (C function), 1620
 (C function), 1622
 nppiMaximumRelativeErrorGetBufferHostSize_32fc_C4R_Ctx (C function), 1622
 (C function), 1622
 nppiMaximumRelativeErrorGetBufferHostSize_32s_C1R_Ctx (C function), 1616
 (C function), 1616
 nppiMaximumRelativeErrorGetBufferHostSize_32s_C2R_Ctx (C function), 1618
 (C function), 1618
 nppiMaximumRelativeErrorGetBufferHostSize_32s_C3R_Ctx (C function), 1620
 (C function), 1620
 nppiMaximumRelativeErrorGetBufferHostSize_32s_C4R_Ctx (C function), 1622
 (C function), 1622
 nppiMaximumRelativeErrorGetBufferHostSize_32sc_C1R_Ctx (C function), 1616
 (C function), 1616
 nppiMaximumRelativeErrorGetBufferHostSize_32sc_C2R_Ctx (C function), 1618
 (C function), 1618
 nppiMaximumRelativeErrorGetBufferHostSize_32sc_C3R_Ctx (C function), 1620
 (C function), 1620
 nppiMaximumRelativeErrorGetBufferHostSize_32sc_C4R_Ctx (C function), 1622
 (C function), 1622
 nppiMaximumRelativeErrorGetBufferHostSize_32u_C1R_Ctx (C function), 1616
 (C function), 1616
 nppiMaximumRelativeErrorGetBufferHostSize_32u_C2R_Ctx (C function), 1618
 (C function), 1618
 nppiMaximumRelativeErrorGetBufferHostSize_32u_C3R_Ctx (C function), 1620
 (C function), 1620
 nppiMaximumRelativeErrorGetBufferHostSize_32u_C4R_Ctx (C function), 1622
 (C function), 1622
 nppiMaximumRelativeErrorGetBufferHostSize_64f_C1R_Ctx (C function), 1617
 (C function), 1617
 nppiMaximumRelativeErrorGetBufferHostSize_64f_C2R_Ctx (C function), 1619
 (C function), 1619
 nppiMaximumRelativeErrorGetBufferHostSize_64f_C3R_Ctx (C function), 1621
 (C function), 1621
 nppiMaximumRelativeErrorGetBufferHostSize_64f_C4R_Ctx (C function), 1623
 (C function), 1623
 nppiMaximumRelativeErrorGetBufferHostSize_8u_AC4R_Ctx (C function), 1379
 (C function), 1379
 nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R_Ctx (C function), 1377
 (C function), 1377
 nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R_Ctx (C function), 1378
 (C function), 1378
 nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R_Ctx (C function), 1378
 (C function), 1378
 nppiMaximumRelativeErrorGetBufferHostSize_16s_AC4R_Ctx (C function), 1379
 (C function), 1379
 nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R_Ctx (C function), 1378
 (C function), 1378
 nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R_Ctx (C function), 1378
 (C function), 1378
 nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R_Ctx (C function), 1378
 (C function), 1378
 nppiMaximumRelativeErrorGetBufferHostSize_16u_AC4R_Ctx (C function), 1379
 (C function), 1379
 nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R_Ctx (C function), 1377
 (C function), 1377
 nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R_Ctx (C function), 1378
 (C function), 1378
 nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R_Ctx (C function), 1378
 (C function), 1378
 nppiMaximumRelativeErrorGetBufferHostSize_32f_AC4R_Ctx (C function), 1379
 (C function), 1379
 nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R_Ctx (C function), 1378
 (C function), 1378
 nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R_Ctx (C function), 1378
 (C function), 1378
 nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R_Ctx (C function), 1379
 (C function), 1379
 nppiMaximumRelativeErrorGetBufferHostSize_8u_AC4R_Ctx (C function), 1381
 (C function), 1381
 nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R_Ctx (C function), 1381
 (C function), 1381

(C function), 1380
 nppiMaxIndxGetBufferHostSize_8u_C3R_Ctx (C function), 1380
 nppiMaxIndxGetBufferHostSize_8u_C4R_Ctx (C function), 1381
 nppiMaxIndxGetBufferHostSize_16s_AC4R_Ctx (C function), 1381
 nppiMaxIndxGetBufferHostSize_16s_C1R_Ctx (C function), 1380
 nppiMaxIndxGetBufferHostSize_16s_C3R_Ctx (C function), 1380
 nppiMaxIndxGetBufferHostSize_16s_C4R_Ctx (C function), 1381
 nppiMaxIndxGetBufferHostSize_16u_AC4R_Ctx (C function), 1381
 nppiMaxIndxGetBufferHostSize_16u_C1R_Ctx (C function), 1380
 nppiMaxIndxGetBufferHostSize_16u_C3R_Ctx (C function), 1380
 nppiMaxIndxGetBufferHostSize_16u_C4R_Ctx (C function), 1381
 nppiMaxIndxGetBufferHostSize_32f_AC4R_Ctx (C function), 1382
 nppiMaxIndxGetBufferHostSize_32f_C1R_Ctx (C function), 1380
 nppiMaxIndxGetBufferHostSize_32f_C3R_Ctx (C function), 1381
 nppiMaxIndxGetBufferHostSize_32f_C4R_Ctx (C function), 1381
 nppiMean_8s_C1MR_Ctx (C function), 1390
 nppiMean_8s_C3CMR_Ctx (C function), 1390
 nppiMean_8u_AC4R_Ctx (C function), 1389
 nppiMean_8u_C1MR_Ctx (C function), 1389
 nppiMean_8u_C1R_Ctx (C function), 1388
 nppiMean_8u_C3CMR_Ctx (C function), 1390
 nppiMean_8u_C3R_Ctx (C function), 1388
 nppiMean_8u_C4R_Ctx (C function), 1388
 nppiMean_16s_AC4R_Ctx (C function), 1389
 nppiMean_16s_C1R_Ctx (C function), 1388
 nppiMean_16s_C3R_Ctx (C function), 1388
 nppiMean_16s_C4R_Ctx (C function), 1389
 nppiMean_16u_AC4R_Ctx (C function), 1389
 nppiMean_16u_C1MR_Ctx (C function), 1390
 nppiMean_16u_C1R_Ctx (C function), 1388
 nppiMean_16u_C3CMR_Ctx (C function), 1390
 nppiMean_16u_C3R_Ctx (C function), 1388
 nppiMean_16u_C4R_Ctx (C function), 1389
 nppiMean_32f_AC4R_Ctx (C function), 1389
 nppiMean_32f_C1MR_Ctx (C function), 1390
 nppiMean_32f_C1R_Ctx (C function), 1388
 nppiMean_32f_C3CMR_Ctx (C function), 1390
 nppiMean_32f_C3R_Ctx (C function), 1388
 nppiMean_32f_C4R_Ctx (C function), 1389
 nppiMean_StdDev_8s_C1MR_Ctx (C function), 1395
 nppiMean_StdDev_8s_C1R_Ctx (C function), 1395
 nppiMean_StdDev_8s_C3CMR_Ctx (C function), 1397
 nppiMean_StdDev_8s_C3CR_Ctx (C function), 1396
 nppiMean_StdDev_8u_C1MR_Ctx (C function), 1395
 nppiMean_StdDev_8u_C1R_Ctx (C function), 1395
 nppiMean_StdDev_8u_C3CMR_Ctx (C function), 1396
 nppiMean_StdDev_8u_C3CR_Ctx (C function), 1396
 nppiMean_StdDev_16u_C1MR_Ctx (C function), 1395
 nppiMean_StdDev_16u_C1R_Ctx (C function), 1395
 nppiMean_StdDev_16u_C3CMR_Ctx (C function), 1397
 nppiMean_StdDev_16u_C3CR_Ctx (C function), 1396
 nppiMean_StdDev_32f_C1MR_Ctx (C function), 1396
 nppiMean_StdDev_32f_C1R_Ctx (C function), 1396
 nppiMean_StdDev_32f_C3CMR_Ctx (C function), 1397
 nppiMean_StdDev_32f_C3CR_Ctx (C function), 1396
 nppiMeanGetBufferHostSize_8s_C1MR_Ctx (C function), 1393
 nppiMeanGetBufferHostSize_8s_C3CMR_Ctx (C function), 1393
 nppiMeanGetBufferHostSize_8u_AC4R_Ctx (C function), 1392
 nppiMeanGetBufferHostSize_8u_C1MR_Ctx (C function), 1392
 nppiMeanGetBufferHostSize_8u_C1R_Ctx (C function), 1391
 nppiMeanGetBufferHostSize_8u_C3CMR_Ctx (C function), 1393
 nppiMeanGetBufferHostSize_8u_C3R_Ctx (C function), 1391
 nppiMeanGetBufferHostSize_8u_C4R_Ctx (C function), 1392
 nppiMeanGetBufferHostSize_16s_AC4R_Ctx (C function), 1392
 nppiMeanGetBufferHostSize_16s_C1R_Ctx (C function), 1391
 nppiMeanGetBufferHostSize_16s_C3R_Ctx (C function), 1391

nppiMeanGetBufferHostSize_16s_C4R_Ctx (C function), 1392	nppiMeanStdDevGetBufferHostSize_32f_C3CMR_Ctx (C function), 1400
nppiMeanGetBufferHostSize_16u_AC4R_Ctx (C function), 1392	nppiMeanStdDevGetBufferHostSize_32f_C3CR_Ctx (C function), 1399
nppiMeanGetBufferHostSize_16u_C1MR_Ctx (C function), 1393	nppiMin_8u_AC4R_Ctx (C function), 1364
nppiMeanGetBufferHostSize_16u_C1R_Ctx (C function), 1391	nppiMin_8u_C1R_Ctx (C function), 1362
nppiMeanGetBufferHostSize_16u_C3CMR_Ctx (C function), 1393	nppiMin_8u_C3R_Ctx (C function), 1363
nppiMeanGetBufferHostSize_16u_C3R_Ctx (C function), 1391	nppiMin_8u_C4R_Ctx (C function), 1363
nppiMeanGetBufferHostSize_16u_C4R_Ctx (C function), 1392	nppiMin_16s_AC4R_Ctx (C function), 1364
nppiMeanGetBufferHostSize_32f_AC4R_Ctx (C function), 1392	nppiMin_16s_C1R_Ctx (C function), 1363
nppiMeanGetBufferHostSize_32f_C1MR_Ctx (C function), 1393	nppiMin_16s_C3R_Ctx (C function), 1363
nppiMeanGetBufferHostSize_32f_C1R_Ctx (C function), 1391	nppiMin_16s_C4R_Ctx (C function), 1364
nppiMeanGetBufferHostSize_32f_C3CMR_Ctx (C function), 1393	nppiMin_16u_AC4R_Ctx (C function), 1364
nppiMeanGetBufferHostSize_32f_C3R_Ctx (C function), 1392	nppiMin_16u_C1R_Ctx (C function), 1362
nppiMeanGetBufferHostSize_32f_C4R_Ctx (C function), 1392	nppiMin_16u_C3R_Ctx (C function), 1363
nppiMeanStdDevGetBufferHostSize_8s_C1MR_Ctx (C function), 1398	nppiMin_16u_C4R_Ctx (C function), 1363
nppiMeanStdDevGetBufferHostSize_8s_C1R_Ctx (C function), 1398	nppiMin_32f_AC4R_Ctx (C function), 1364
nppiMeanStdDevGetBufferHostSize_8s_C3CMR_Ctx (C function), 1400	nppiMin_32f_C1R_Ctx (C function), 1363
nppiMeanStdDevGetBufferHostSize_8s_C3CR_Ctx (C function), 1399	nppiMin_32f_C3R_Ctx (C function), 1363
nppiMeanStdDevGetBufferHostSize_8u_C1MR_Ctx (C function), 1398	nppiMin_32f_C4R_Ctx (C function), 1364
nppiMeanStdDevGetBufferHostSize_8u_C1R_Ctx (C function), 1398	nppiMinEvery_8u_AC4IR_Ctx (C function), 1483
nppiMeanStdDevGetBufferHostSize_8u_C3CMR_Ctx (C function), 1399	nppiMinEvery_8u_C1IR_Ctx (C function), 1481
nppiMeanStdDevGetBufferHostSize_8u_C3CR_Ctx (C function), 1399	nppiMinEvery_8u_C3IR_Ctx (C function), 1482
nppiMeanStdDevGetBufferHostSize_16u_C1MR_Ctx (C function), 1399	nppiMinEvery_8u_C4IR_Ctx (C function), 1482
nppiMeanStdDevGetBufferHostSize_16u_C1R_Ctx (C function), 1398	nppiMinEvery_16s_AC4IR_Ctx (C function), 1483
nppiMeanStdDevGetBufferHostSize_16u_C3CMR_Ctx (C function), 1400	nppiMinEvery_16s_C1IR_Ctx (C function),
nppiMeanStdDevGetBufferHostSize_16u_C3CR_Ctx (C function), 1399	nppiMinEvery_16s_C3IR_Ctx (C function),
nppiMeanStdDevGetBufferHostSize_32f_C1MR_Ctx (C function), 1399	nppiMinEvery_16s_C4IR_Ctx (C function),
nppiMeanStdDevGetBufferHostSize_32f_C1R_Ctx (C function), 1398	nppiMinEvery_16u_AC4IR_Ctx (C function),
nppiMeanStdDevGetBufferHostSize_32f_C3R_Ctx (C function), 1398	nppiMinEvery_16u_C1IR_Ctx (C function),
nppiMeanStdDevGetBufferHostSize_32f_C4R_Ctx (C function), 1398	nppiMinEvery_16u_C3IR_Ctx (C function),
	nppiMinEvery_16u_C4IR_Ctx (C function),
	nppiMinEvery_32f_AC4IR_Ctx (C function),
	nppiMinEvery_32f_C1IR_Ctx (C function),
	nppiMinEvery_32f_C3IR_Ctx (C function),
	nppiMinEvery_32f_C4IR_Ctx (C function),
	nppiMinGetBufferHostSize_8u_AC4R_Ctx (C function), 1366
	nppiMinGetBufferHostSize_8u_C1R_Ctx (C function), 1365
	nppiMinGetBufferHostSize_8u_C3R_Ctx (C

function), 1365
 nppiMinGetBufferHostSize_8u_C4R_Ctx (C function), 1366
 nppiMinGetBufferHostSize_16s_AC4R_Ctx (C function), 1366
 nppiMinGetBufferHostSize_16s_C1R_Ctx (C function), 1365
 nppiMinGetBufferHostSize_16s_C3R_Ctx (C function), 1365
 nppiMinGetBufferHostSize_16s_C4R_Ctx (C function), 1366
 nppiMinGetBufferHostSize_16u_AC4R_Ctx (C function), 1366
 nppiMinGetBufferHostSize_16u_C1R_Ctx (C function), 1365
 nppiMinGetBufferHostSize_16u_C3R_Ctx (C function), 1365
 nppiMinGetBufferHostSize_16u_C4R_Ctx (C function), 1366
 nppiMinGetBufferHostSize_32f_AC4R_Ctx (C function), 1366
 nppiMinGetBufferHostSize_32f_C1R_Ctx (C function), 1365
 nppiMinGetBufferHostSize_32f_C3R_Ctx (C function), 1365
 nppiMinGetBufferHostSize_32f_C4R_Ctx (C function), 1366
 nppiMinIndx_8u_AC4R_Ctx (C function), 1369
 nppiMinIndx_8u_C1R_Ctx (C function), 1367
 nppiMinIndx_8u_C3R_Ctx (C function), 1368
 nppiMinIndx_8u_C4R_Ctx (C function), 1368
 nppiMinIndx_16s_AC4R_Ctx (C function), 1369
 nppiMinIndx_16s_C1R_Ctx (C function), 1368
 nppiMinIndx_16s_C3R_Ctx (C function), 1368
 nppiMinIndx_16s_C4R_Ctx (C function), 1368
 nppiMinIndx_16u_AC4R_Ctx (C function), 1369
 nppiMinIndx_16u_C1R_Ctx (C function), 1367
 nppiMinIndx_16u_C3R_Ctx (C function), 1368
 nppiMinIndx_16u_C4R_Ctx (C function), 1368
 nppiMinIndx_32f_AC4R_Ctx (C function), 1369
 nppiMinIndx_32f_C1R_Ctx (C function), 1368
 nppiMinIndx_32f_C3R_Ctx (C function), 1368
 nppiMinIndx_32f_C4R_Ctx (C function), 1369
 nppiMinIndxGetBufferHostSize_8u_AC4R_Ctx (C function), 1371
 nppiMinIndxGetBufferHostSize_8u_C1R_Ctx (C function), 1370
 nppiMinIndxGetBufferHostSize_8u_C3R_Ctx (C function), 1370
 nppiMinIndxGetBufferHostSize_8u_C4R_Ctx (C function), 1371
 nppiMinIndxGetBufferHostSize_16s_AC4R_Ctx (C function), 1371
 nppiMinIndxGetBufferHostSize_16s_C1R_Ctx (C function), 1371
 nppiMinIndxGetBufferHostSize_16s_C3R_Ctx (C function), 1370
 nppiMinIndxGetBufferHostSize_16s_C4R_Ctx (C function), 1371
 nppiMinIndxGetBufferHostSize_16u_AC4R_Ctx (C function), 1371
 nppiMinIndxGetBufferHostSize_16u_C1R_Ctx (C function), 1370
 nppiMinIndxGetBufferHostSize_16u_C3R_Ctx (C function), 1370
 nppiMinIndxGetBufferHostSize_16u_C4R_Ctx (C function), 1371
 nppiMinIndxGetBufferHostSize_32f_AC4R_Ctx (C function), 1372
 nppiMinIndxGetBufferHostSize_32f_C1R_Ctx (C function), 1370
 nppiMinIndxGetBufferHostSize_32f_C3R_Ctx (C function), 1371
 nppiMinIndxGetBufferHostSize_32f_C4R_Ctx (C function), 1371
 nppiMinMax_8u_AC4R_Ctx (C function), 1383
 nppiMinMax_8u_C1R_Ctx (C function), 1382
 nppiMinMax_8u_C3R_Ctx (C function), 1383
 nppiMinMax_8u_C4R_Ctx (C function), 1384
 nppiMinMax_16s_AC4R_Ctx (C function), 1384
 nppiMinMax_16s_C1R_Ctx (C function), 1383
 nppiMinMax_16s_C3R_Ctx (C function), 1383
 nppiMinMax_16s_C4R_Ctx (C function), 1384
 nppiMinMax_16u_AC4R_Ctx (C function), 1383
 nppiMinMax_16u_C1R_Ctx (C function), 1382
 nppiMinMax_16u_C3R_Ctx (C function), 1383
 nppiMinMax_16u_C4R_Ctx (C function), 1384
 nppiMinMax_32f_AC4R_Ctx (C function), 1384
 nppiMinMax_32f_C1R_Ctx (C function), 1383
 nppiMinMax_32f_C3R_Ctx (C function), 1383
 nppiMinMax_32f_C4R_Ctx (C function), 1384
 nppiMinMaxGetBufferHostSize_8u_AC4R_Ctx (C function), 1386
 nppiMinMaxGetBufferHostSize_8u_C1R_Ctx (C function), 1385
 nppiMinMaxGetBufferHostSize_8u_C3R_Ctx (C function), 1385
 nppiMinMaxGetBufferHostSize_8u_C4R_Ctx (C function), 1386
 nppiMinMaxGetBufferHostSize_16s_AC4R_Ctx (C function), 1386
 nppiMinMaxGetBufferHostSize_16s_C1R_Ctx (C function), 1385
 nppiMinMaxGetBufferHostSize_16s_C3R_Ctx (C function), 1385
 nppiMinMaxGetBufferHostSize_16s_C4R_Ctx (C function), 1386
 nppiMinMaxGetBufferHostSize_16u_AC4R_Ctx (C function), 1386

(C function), 1386
 nppiMinMaxGetBufferHostSize_16u_C1R_Ctx (C function), 1385
 nppiMinMaxGetBufferHostSize_16u_C3R_Ctx (C function), 1385
 nppiMinMaxGetBufferHostSize_16u_C4R_Ctx (C function), 1386
 nppiMinMaxGetBufferHostSize_32f_AC4R_Ctx (C function), 1386
 nppiMinMaxGetBufferHostSize_32f_C1R_Ctx (C function), 1385
 nppiMinMaxGetBufferHostSize_32f_C3R_Ctx (C function), 1386
 nppiMinMaxGetBufferHostSize_32f_C4R_Ctx (C function), 1386
 nppiMirror_8u_AC4IR_Ctx (C function), 1261
 nppiMirror_8u_AC4R_Ctx (C function), 1261
 nppiMirror_8u_C1IR_Ctx (C function), 1261
 nppiMirror_8u_C1R_Ctx (C function), 1261
 nppiMirror_8u_C3IR_Ctx (C function), 1261
 nppiMirror_8u_C3R_Ctx (C function), 1261
 nppiMirror_8u_C4IR_Ctx (C function), 1261
 nppiMirror_8u_C4R_Ctx (C function), 1261
 nppiMirror_16s_AC4IR_Ctx (C function), 1263
 nppiMirror_16s_AC4R_Ctx (C function), 1263
 nppiMirror_16s_C1IR_Ctx (C function), 1262
 nppiMirror_16s_C1R_Ctx (C function), 1262
 nppiMirror_16s_C3IR_Ctx (C function), 1263
 nppiMirror_16s_C3R_Ctx (C function), 1263
 nppiMirror_16s_C4IR_Ctx (C function), 1263
 nppiMirror_16s_C4R_Ctx (C function), 1263
 nppiMirror_16u_AC4IR_Ctx (C function), 1262
 nppiMirror_16u_AC4R_Ctx (C function), 1262
 nppiMirror_16u_C1IR_Ctx (C function), 1262
 nppiMirror_16u_C1R_Ctx (C function), 1261
 nppiMirror_16u_C3IR_Ctx (C function), 1262
 nppiMirror_16u_C3R_Ctx (C function), 1262
 nppiMirror_16u_C4IR_Ctx (C function), 1262
 nppiMirror_16u_C4R_Ctx (C function), 1262
 nppiMirror_32f_AC4IR_Ctx (C function), 1265
 nppiMirror_32f_AC4R_Ctx (C function), 1265
 nppiMirror_32f_C1IR_Ctx (C function), 1264
 nppiMirror_32f_C1R_Ctx (C function), 1264
 nppiMirror_32f_C3IR_Ctx (C function), 1264
 nppiMirror_32f_C3R_Ctx (C function), 1264
 nppiMirror_32f_C4IR_Ctx (C function), 1265
 nppiMirror_32f_C4R_Ctx (C function), 1265
 nppiMirror_32s_AC4IR_Ctx (C function), 1264
 nppiMirror_32s_AC4R_Ctx (C function), 1264
 nppiMirror_32s_C1IR_Ctx (C function), 1263
 nppiMirror_32s_C1R_Ctx (C function), 1263
 nppiMirror_32s_C3IR_Ctx (C function), 1264
 nppiMirror_32s_C3R_Ctx (C function), 1263
 nppiMirror_32s_C4IR_Ctx (C function), 1264
 nppiMirror_32s_C4R_Ctx (C function), 1264
 nppiMirror_32s_C4R_Ctx (C function), 1264
 NppiMirrorBatchCXR (C struct), 49
 NppiMirrorBatchCXR.nDstStep (C var), 50
 NppiMirrorBatchCXR.nSrcStep (C var), 50
 NppiMirrorBatchCXR.pDst (C var), 50
 NppiMirrorBatchCXR.pSrc (C var), 50
 nppiMorphBlackHatBorder_8u_C1R_Ctx (C function), 1351
 nppiMorphBlackHatBorder_8u_C3R_Ctx (C function), 1351
 nppiMorphBlackHatBorder_8u_C4R_Ctx (C function), 1351
 nppiMorphBlackHatBorder_16s_C1R_Ctx (C function), 1352
 nppiMorphBlackHatBorder_16u_C1R_Ctx (C function), 1352
 nppiMorphBlackHatBorder_32f_C1R_Ctx (C function), 1352
 nppiMorphBlackHatBorder_32f_C3R_Ctx (C function), 1352
 nppiMorphBlackHatBorder_32f_C4R_Ctx (C function), 1352
 nppiMorphCloseBorder_8u_C1R_Ctx (C function), 1342
 nppiMorphCloseBorder_8u_C3R_Ctx (C function), 1342
 nppiMorphCloseBorder_8u_C4R_Ctx (C function), 1342
 nppiMorphCloseBorder_16s_C1R_Ctx (C function), 1343
 nppiMorphCloseBorder_16u_C1R_Ctx (C function), 1343
 nppiMorphCloseBorder_32f_C1R_Ctx (C function), 1343
 nppiMorphCloseBorder_32f_C3R_Ctx (C function), 1343
 nppiMorphCloseBorder_32f_C4R_Ctx (C function), 1344
 nppiMorphGetBufferSize_8u_C1R (C function), 1340
 nppiMorphGetBufferSize_8u_C3R (C function), 1340
 nppiMorphGetBufferSize_8u_C4R (C function), 1340
 nppiMorphGetBufferSize_16s_C1R (C function), 1340
 nppiMorphGetBufferSize_16u_C1R (C function), 1340
 nppiMorphGetBufferSize_32f_C1R (C function), 1340
 nppiMorphGetBufferSize_32f_C3R (C function), 1341
 nppiMorphGetBufferSize_32f_C4R (C function), 1341

nppiMorphGradientBorder_8u_C1R_Ctx (C function), 1354
 nppiMorphGradientBorder_8u_C3R_Ctx (C function), 1354
 nppiMorphGradientBorder_8u_C4R_Ctx (C function), 1354
 nppiMorphGradientBorder_16s_C1R_Ctx (C function), 1355
 nppiMorphGradientBorder_16u_C1R_Ctx (C function), 1355
 nppiMorphGradientBorder_32f_C1R_Ctx (C function), 1355
 nppiMorphGradientBorder_32f_C3R_Ctx (C function), 1355
 nppiMorphGradientBorder_32f_C4R_Ctx (C function), 1355
 nppiMorphOpenBorder_8u_C1R_Ctx (C function), 1345
 nppiMorphOpenBorder_8u_C3R_Ctx (C function), 1345
 nppiMorphOpenBorder_8u_C4R_Ctx (C function), 1345
 nppiMorphOpenBorder_16s_C1R_Ctx (C function), 1346
 nppiMorphOpenBorder_16u_C1R_Ctx (C function), 1346
 nppiMorphOpenBorder_32f_C1R_Ctx (C function), 1346
 nppiMorphOpenBorder_32f_C3R_Ctx (C function), 1346
 nppiMorphOpenBorder_32f_C4R_Ctx (C function), 1346
 nppiMorphTopHatBorder_8u_C1R_Ctx (C function), 1348
 nppiMorphTopHatBorder_8u_C3R_Ctx (C function), 1348
 nppiMorphTopHatBorder_8u_C4R_Ctx (C function), 1348
 nppiMorphTopHatBorder_16s_C1R_Ctx (C function), 1349
 nppiMorphTopHatBorder_16u_C1R_Ctx (C function), 1349
 nppiMorphTopHatBorder_32f_C1R_Ctx (C function), 1349
 nppiMorphTopHatBorder_32f_C3R_Ctx (C function), 1349
 nppiMorphTopHatBorder_32f_C4R_Ctx (C function), 1349
 nppiMSE_8u_C1R_Ctx (C function), 1640
 nppiMSE_8u_C3R_Ctx (C function), 1640
 nppiMSEBatch_8u_C1R_Advanced_Ctx (C function), 1654
 nppiMSEBatch_8u_C1R_Ctx (C function), 1646
 nppiMSEBatch_8u_C3R_Advanced_Ctx (C function), 1654
 nppiMSEBatch_8u_C3R_Ctx (C function), 1647
 nppiMSEBatchGetBufferHostSize_8u_C1R_Ctx (C function), 1652
 nppiMSEBatchGetBufferHostSize_8u_C3R_Ctx (C function), 1652
 nppiMSEGetBufferHostSize_8u_C1R_Ctx (C function), 1645
 nppiMSEGetBufferHostSize_8u_C3R_Ctx (C function), 1645
 nppiMSSSIM_8u_C1R_Ctx (C function), 1643
 nppiMSSSIMGetBufferHostSize_8u_C1R_Ctx (C function), 1645
 nppiMul_8u_AC4IRSfs_Ctx (C function), 269
 nppiMul_8u_AC4RSfs_Ctx (C function), 269
 nppiMul_8u_C1IRSfs_Ctx (C function), 267
 nppiMul_8u_C1RSfs_Ctx (C function), 267
 nppiMul_8u_C3IRSfs_Ctx (C function), 268
 nppiMul_8u_C3RSfs_Ctx (C function), 268
 nppiMul_8u_C4IRSfs_Ctx (C function), 270
 nppiMul_8u_C4RSfs_Ctx (C function), 270
 nppiMul_16f_C1IR_Ctx (C function), 285
 nppiMul_16f_C1R_Ctx (C function), 285
 nppiMul_16f_C3IR_Ctx (C function), 286
 nppiMul_16f_C3R_Ctx (C function), 286
 nppiMul_16f_C4IR_Ctx (C function), 287
 nppiMul_16f_C4R_Ctx (C function), 286
 nppiMul_16s_AC4IRSfs_Ctx (C function), 276
 nppiMul_16s_AC4RSfs_Ctx (C function), 276
 nppiMul_16s_C1IRSfs_Ctx (C function), 274
 nppiMul_16s_C1RSfs_Ctx (C function), 274
 nppiMul_16s_C3IRSfs_Ctx (C function), 275
 nppiMul_16s_C3RSfs_Ctx (C function), 275
 nppiMul_16s_C4IRSfs_Ctx (C function), 277
 nppiMul_16s_C4RSfs_Ctx (C function), 277
 nppiMul_16sc_AC4IRSfs_Ctx (C function), 280
 nppiMul_16sc_AC4RSfs_Ctx (C function), 279
 nppiMul_16sc_C1IRSfs_Ctx (C function), 278
 nppiMul_16sc_C1RSfs_Ctx (C function), 277
 nppiMul_16sc_C3IRSfs_Ctx (C function), 279
 nppiMul_16sc_C3RSfs_Ctx (C function), 278
 nppiMul_16u_AC4IRSfs_Ctx (C function), 273
 nppiMul_16u_AC4RSfs_Ctx (C function), 272
 nppiMul_16u_C1IRSfs_Ctx (C function), 271
 nppiMul_16u_C1RSfs_Ctx (C function), 270
 nppiMul_16u_C3IRSfs_Ctx (C function), 272
 nppiMul_16u_C3RSfs_Ctx (C function), 271
 nppiMul_16u_C4IRSfs_Ctx (C function), 273
 nppiMul_16u_C4RSfs_Ctx (C function), 273
 nppiMul_32f_AC4IR_Ctx (C function), 289
 nppiMul_32f_AC4R_Ctx (C function), 289
 nppiMul_32f_C1IR_Ctx (C function), 288
 nppiMul_32f_C1R_Ctx (C function), 287
 nppiMul_32f_C3IR_Ctx (C function), 288

nppiMul_32f_C3R_Ctx (C function), 288
 nppiMul_32f_C4IR_Ctx (C function), 290
 nppiMul_32f_C4R_Ctx (C function), 289
 nppiMul_32fc_AC4IR_Ctx (C function), 292
 nppiMul_32fc_AC4R_Ctx (C function), 292
 nppiMul_32fc_C1IR_Ctx (C function), 291
 nppiMul_32fc_C1R_Ctx (C function), 290
 nppiMul_32fc_C3IR_Ctx (C function), 291
 nppiMul_32fc_C3R_Ctx (C function), 291
 nppiMul_32fc_C4IR_Ctx (C function), 293
 nppiMul_32fc_C4R_Ctx (C function), 292
 nppiMul_32s_C1IRSfs_Ctx (C function), 281
 nppiMul_32s_C1R_Ctx (C function), 280
 nppiMul_32s_C1RSfs_Ctx (C function), 280
 nppiMul_32s_C3IRSfs_Ctx (C function), 282
 nppiMul_32s_C3RSfs_Ctx (C function), 281
 nppiMul_32sc_AC4IRSfs_Ctx (C function), 284
 nppiMul_32sc_AC4RSfs_Ctx (C function), 284
 nppiMul_32sc_C1IRSfs_Ctx (C function), 283
 nppiMul_32sc_C1RSfs_Ctx (C function), 282
 nppiMul_32sc_C3IRSfs_Ctx (C function), 284
 nppiMul_32sc_C3RSfs_Ctx (C function), 283
 nppiMulC_8u_AC4IRSfs_Ctx (C function), 97
 nppiMulC_8u_AC4RSfs_Ctx (C function), 96
 nppiMulC_8u_C1IRSfs_Ctx (C function), 94
 nppiMulC_8u_C1RSfs_Ctx (C function), 93
 nppiMulC_8u_C3IRSfs_Ctx (C function), 95
 nppiMulC_8u_C3RSfs_Ctx (C function), 94
 nppiMulC_8u_C4IRSfs_Ctx (C function), 99
 nppiMulC_8u_C4RSfs_Ctx (C function), 98
 nppiMulC_16f_C1IR_Ctx (C function), 121
 nppiMulC_16f_C1R_Ctx (C function), 121
 nppiMulC_16f_C3IR_Ctx (C function), 123
 nppiMulC_16f_C3R_Ctx (C function), 122
 nppiMulC_16f_C4IR_Ctx (C function), 124
 nppiMulC_16f_C4R_Ctx (C function), 124
 nppiMulC_16s_AC4IRSfs_Ctx (C function), 110
 nppiMulC_16s_AC4RSfs_Ctx (C function), 109
 nppiMulC_16s_C1IRSfs_Ctx (C function), 107
 nppiMulC_16s_C1RSfs_Ctx (C function), 106
 nppiMulC_16s_C3IRSfs_Ctx (C function), 109
 nppiMulC_16s_C3RSfs_Ctx (C function), 108
 nppiMulC_16s_C4IRSfs_Ctx (C function), 112
 nppiMulC_16s_C4RSfs_Ctx (C function), 111
 nppiMulC_16sc_AC4IRSfs_Ctx (C function), 115
 nppiMulC_16sc_AC4RSfs_Ctx (C function), 114
 nppiMulC_16sc_C1IRSfs_Ctx (C function), 113
 nppiMulC_16sc_C1RSfs_Ctx (C function), 113
 nppiMulC_16sc_C3IRSfs_Ctx (C function), 114
 nppiMulC_16sc_C3RSfs_Ctx (C function), 113
 nppiMulC_16u_AC4IRSfs_Ctx (C function), 104
 nppiMulC_16u_AC4RSfs_Ctx (C function), 103
 nppiMulC_16u_C1IRSfs_Ctx (C function), 100
 nppiMulC_16u_C1RSfs_Ctx (C function), 99
 nppiMulC_16u_C3IRSfs_Ctx (C function), 102
 nppiMulC_16u_C3RSfs_Ctx (C function), 101
 nppiMulC_16u_C4IRSfs_Ctx (C function), 105
 nppiMulC_16u_C4RSfs_Ctx (C function), 104
 nppiMulC_32f_AC4IR_Ctx (C function), 129
 nppiMulC_32f_AC4R_Ctx (C function), 128
 nppiMulC_32f_C1IR_Ctx (C function), 126
 nppiMulC_32f_C1R_Ctx (C function), 125
 nppiMulC_32f_C3IR_Ctx (C function), 127
 nppiMulC_32f_C3R_Ctx (C function), 126
 nppiMulC_32f_C4IR_Ctx (C function), 130
 nppiMulC_32f_C4R_Ctx (C function), 129
 nppiMulC_32fc_AC4IR_Ctx (C function), 133
 nppiMulC_32fc_AC4R_Ctx (C function), 132
 nppiMulC_32fc_C1IR_Ctx (C function), 131
 nppiMulC_32fc_C1R_Ctx (C function), 131
 nppiMulC_32fc_C3IR_Ctx (C function), 132
 nppiMulC_32fc_C3R_Ctx (C function), 131
 nppiMulC_32fc_C4IR_Ctx (C function), 133
 nppiMulC_32fc_C4R_Ctx (C function), 133
 nppiMulC_32s_C1IRSfs_Ctx (C function), 116
 nppiMulC_32s_C1RSfs_Ctx (C function), 115
 nppiMulC_32s_C3IRSfs_Ctx (C function), 117
 nppiMulC_32s_C3RSfs_Ctx (C function), 117
 nppiMulC_32sc_AC4IRSfs_Ctx (C function), 120
 nppiMulC_32sc_AC4RSfs_Ctx (C function), 120
 nppiMulC_32sc_C1IRSfs_Ctx (C function), 119
 nppiMulC_32sc_C1RSfs_Ctx (C function), 118
 nppiMulC_32sc_C3IRSfs_Ctx (C function), 119
 nppiMulC_32sc_C3RSfs_Ctx (C function), 119
 nppiMulCScale_8u_AC4IR_Ctx (C function), 138
 nppiMulCScale_8u_AC4R_Ctx (C function), 137
 nppiMulCScale_8u_C1IR_Ctx (C function), 135
 nppiMulCScale_8u_C1R_Ctx (C function), 134
 nppiMulCScale_8u_C3IR_Ctx (C function), 136
 nppiMulCScale_8u_C3R_Ctx (C function), 135
 nppiMulCScale_8u_C4IR_Ctx (C function), 139
 nppiMulCScale_8u_C4R_Ctx (C function), 138
 nppiMulCScale_16u_AC4IR_Ctx (C function), 144
 nppiMulCScale_16u_AC4R_Ctx (C function), 143
 nppiMulCScale_16u_C1IR_Ctx (C function), 141
 nppiMulCScale_16u_C1R_Ctx (C function), 140
 nppiMulCScale_16u_C3IR_Ctx (C function), 142
 nppiMulCScale_16u_C3R_Ctx (C function), 141
 nppiMulCScale_16u_C4IR_Ctx (C function), 145
 nppiMulCScale_16u_C4R_Ctx (C function), 144

nppiMulDeviceC_8u_AC4IRSfs_Ctx (C function), 97	nppiMulDeviceC_16u_C3RSfs_Ctx (C function), 101
nppiMulDeviceC_8u_AC4RSfs_Ctx (C function), 97	nppiMulDeviceC_16u_C4IRSfs_Ctx (C function), 106
nppiMulDeviceC_8u_C1IRSfs_Ctx (C function), 94	nppiMulDeviceC_16u_C4RSfs_Ctx (C function), 105
nppiMulDeviceC_8u_C1RSfs_Ctx (C function), 93	nppiMulDeviceC_32f_AC4IR_Ctx (C function), 129
nppiMulDeviceC_8u_C3IRSfs_Ctx (C function), 96	nppiMulDeviceC_32f_AC4R_Ctx (C function), 128
nppiMulDeviceC_8u_C3RSfs_Ctx (C function), 95	nppiMulDeviceC_32f_C1IR_Ctx (C function), 126
nppiMulDeviceC_8u_C4IRSfs_Ctx (C function), 99	nppiMulDeviceC_32f_C1R_Ctx (C function), 125
nppiMulDeviceC_8u_C4RSfs_Ctx (C function), 98	nppiMulDeviceC_32f_C3IR_Ctx (C function), 127
nppiMulDeviceC_16f_C1IR_Ctx (C function), 122	nppiMulDeviceC_32f_C3R_Ctx (C function), 127
nppiMulDeviceC_16f_C1R_Ctx (C function), 121	nppiMulDeviceC_32f_C4IR_Ctx (C function), 130
nppiMulDeviceC_16f_C3IR_Ctx (C function), 123	nppiMulDeviceC_32f_C4R_Ctx (C function), 130
nppiMulDeviceC_16f_C3R_Ctx (C function), 123	nppiMulDeviceC_32s_C1IRSfs_Ctx (C function), 116
nppiMulDeviceC_16f_C4IR_Ctx (C function), 125	nppiMulDeviceC_32s_C1RSfs_Ctx (C function), 115
nppiMulDeviceC_16f_C4R_Ctx (C function), 124	nppiMulDeviceC_32s_C3IRSfs_Ctx (C function), 118
nppiMulDeviceC_16s_AC4IRSfs_Ctx (C function), 111	nppiMulDeviceC_32s_C3RSfs_Ctx (C function), 117
nppiMulDeviceC_16s_AC4RSfs_Ctx (C function), 110	nppiMulDeviceC_Scale_8u_AC4IR_Ctx (C function), 138
nppiMulDeviceC_16s_C1IRSfs_Ctx (C function), 107	nppiMulDeviceC_Scale_8u_AC4R_Ctx (C function), 137
nppiMulDeviceC_16s_C1RSfs_Ctx (C function), 106	nppiMulDeviceC_Scale_8u_C1IR_Ctx (C function), 135
nppiMulDeviceC_16s_C3IRSfs_Ctx (C function), 109	nppiMulDeviceC_Scale_8u_C1R_Ctx (C function), 134
nppiMulDeviceC_16s_C3RSfs_Ctx (C function), 108	nppiMulDeviceC_Scale_8u_C3IR_Ctx (C function), 136
nppiMulDeviceC_16s_C4IRSfs_Ctx (C function), 112	nppiMulDeviceC_Scale_8u_C3R_Ctx (C function), 136
nppiMulDeviceC_16s_C4RSfs_Ctx (C function), 111	nppiMulDeviceC_Scale_8u_C4IR_Ctx (C function), 140
nppiMulDeviceC_16u_AC4IRSfs_Ctx (C function), 104	nppiMulDeviceC_Scale_8u_C4R_Ctx (C function), 139
nppiMulDeviceC_16u_AC4RSfs_Ctx (C function), 103	nppiMulDeviceC_Scale_16u_AC4IR_Ctx (C function), 144
nppiMulDeviceC_16u_C1IRSfs_Ctx (C function), 101	nppiMulDeviceC_Scale_16u_AC4R_Ctx (C function), 143
nppiMulDeviceC_16u_C1RSfs_Ctx (C function), 100	nppiMulDeviceC_Scale_16u_C1IR_Ctx (C function), 141
nppiMulDeviceC_16u_C3IRSfs_Ctx (C function), 102	nppiMulDeviceC_Scale_16u_C1R_Ctx (C function), 140

nppiMulDeviceCScale_16u_C3IR_Ctx (C function), 142
 nppiMulDeviceCScale_16u_C3R_Ctx (C function), 142
 nppiMulDeviceCScale_16u_C4IR_Ctx (C function), 146
 nppiMulDeviceCScale_16u_C4R_Ctx (C function), 145
 nppiMulScale_8u_AC4IR_Ctx (C function), 296
 nppiMulScale_8u_AC4R_Ctx (C function), 295
 nppiMulScale_8u_C1IR_Ctx (C function), 294
 nppiMulScale_8u_C1R_Ctx (C function), 294
 nppiMulScale_8u_C3IR_Ctx (C function), 295
 nppiMulScale_8u_C3R_Ctx (C function), 294
 nppiMulScale_8u_C4IR_Ctx (C function), 296
 nppiMulScale_8u_C4R_Ctx (C function), 296
 nppiMulScale_16u_AC4IR_Ctx (C function), 299
 nppiMulScale_16u_AC4R_Ctx (C function), 298
 nppiMulScale_16u_C1IR_Ctx (C function), 297
 nppiMulScale_16u_C1R_Ctx (C function), 297
 nppiMulScale_16u_C3IR_Ctx (C function), 298
 nppiMulScale_16u_C3R_Ctx (C function), 297
 nppiMulScale_16u_C4IR_Ctx (C function), 299
 nppiMulScale_16u_C4R_Ctx (C function), 299
 NppiNorm (C enum), 41
 nppiNorm_Inf_8s_C1MR_Ctx (C function), 1405
 nppiNorm_Inf_8s_C3CMR_Ctx (C function), 1405
 nppiNorm_Inf_8u_AC4R_Ctx (C function), 1404
 nppiNorm_Inf_8u_C1MR_Ctx (C function), 1405
 nppiNorm_Inf_8u_C1R_Ctx (C function), 1403
 nppiNorm_Inf_8u_C3CMR_Ctx (C function), 1405
 nppiNorm_Inf_8u_C3R_Ctx (C function), 1403
 nppiNorm_Inf_8u_C4R_Ctx (C function), 1404
 nppiNorm_Inf_16s_AC4R_Ctx (C function), 1404
 nppiNorm_Inf_16s_C1R_Ctx (C function), 1403
 nppiNorm_Inf_16s_C3R_Ctx (C function), 1403
 nppiNorm_Inf_16s_C4R_Ctx (C function), 1404
 nppiNorm_Inf_16u_AC4R_Ctx (C function), 1404
 nppiNorm_Inf_16u_C1MR_Ctx (C function), 1405
 nppiNorm_Inf_16u_C1R_Ctx (C function), 1403
 nppiNorm_Inf_16u_C3CMR_Ctx (C function), 1405
 nppiNorm_Inf_16u_C3R_Ctx (C function), 1403
 nppiNorm_Inf_16u_C4R_Ctx (C function), 1404
 nppiNorm_Inf_32f_AC4R_Ctx (C function), 1404
 nppiNorm_Inf_32f_C1MR_Ctx (C function), 1405
 nppiNorm_Inf_32f_C1R_Ctx (C function), 1403
 nppiNorm_Inf_32f_C3CMR_Ctx (C function), 1405
 nppiNorm_Inf_32f_C3R_Ctx (C function), 1403
 nppiNorm_Inf_32f_C4R_Ctx (C function), 1404
 nppiNorm_Inf_32s_C1R_Ctx (C function), 1403
 nppiNorm_L1_8s_C1MR_Ctx (C function), 1411
 nppiNorm_L1_8s_C3CMR_Ctx (C function), 1412
 nppiNorm_L1_8u_AC4R_Ctx (C function), 1410
 nppiNorm_L1_8u_C1MR_Ctx (C function), 1411
 nppiNorm_L1_8u_C1R_Ctx (C function), 1409
 nppiNorm_L1_8u_C3CMR_Ctx (C function), 1412
 nppiNorm_L1_8u_C3R_Ctx (C function), 1410
 nppiNorm_L1_8u_C4R_Ctx (C function), 1411
 nppiNorm_L1_16s_AC4R_Ctx (C function), 1410
 nppiNorm_L1_16s_C1R_Ctx (C function), 1409
 nppiNorm_L1_16s_C3R_Ctx (C function), 1410
 nppiNorm_L1_16s_C4R_Ctx (C function), 1411
 nppiNorm_L1_16u_AC4R_Ctx (C function), 1410
 nppiNorm_L1_16u_C1MR_Ctx (C function), 1411
 nppiNorm_L1_16u_C1R_Ctx (C function), 1409
 nppiNorm_L1_16u_C3CMR_Ctx (C function), 1412
 nppiNorm_L1_16u_C3R_Ctx (C function), 1410
 nppiNorm_L1_16u_C4R_Ctx (C function), 1411
 nppiNorm_L1_32f_AC4R_Ctx (C function), 1410
 nppiNorm_L1_32f_C1MR_Ctx (C function), 1411
 nppiNorm_L1_32f_C1R_Ctx (C function), 1410
 nppiNorm_L1_32f_C3CMR_Ctx (C function), 1412
 nppiNorm_L1_32f_C3R_Ctx (C function), 1410
 nppiNorm_L1_32f_C4R_Ctx (C function), 1411
 nppiNorm_L2_8s_C1MR_Ctx (C function), 1418
 nppiNorm_L2_8s_C3CMR_Ctx (C function), 1418
 nppiNorm_L2_8u_AC4R_Ctx (C function), 1417
 nppiNorm_L2_8u_C1MR_Ctx (C function), 1418
 nppiNorm_L2_8u_C1R_Ctx (C function), 1416
 nppiNorm_L2_8u_C3CMR_Ctx (C function), 1418
 nppiNorm_L2_8u_C3R_Ctx (C function), 1416
 nppiNorm_L2_8u_C4R_Ctx (C function), 1417
 nppiNorm_L2_16s_AC4R_Ctx (C function), 1417
 nppiNorm_L2_16s_C1R_Ctx (C function), 1416
 nppiNorm_L2_16s_C3R_Ctx (C function), 1416
 nppiNorm_L2_16s_C4R_Ctx (C function), 1417
 nppiNorm_L2_16u_AC4R_Ctx (C function), 1417
 nppiNorm_L2_16u_C1MR_Ctx (C function), 1418
 nppiNorm_L2_16u_C1R_Ctx (C function), 1416
 nppiNorm_L2_16u_C3CMR_Ctx (C function), 1418
 nppiNorm_L2_16u_C3R_Ctx (C function), 1416
 nppiNorm_L2_16u_C4R_Ctx (C function), 1417
 nppiNorm_L2_32f_AC4R_Ctx (C function), 1417
 nppiNorm_L2_32f_C1MR_Ctx (C function), 1418
 nppiNorm_L2_32f_C1R_Ctx (C function), 1416

nppiNorm_L2_32f_C3CMR_Ctx (C function), 1418	nppiNormDiff_L1_8s_C3CMR_Ctx (C function), 1433
nppiNorm_L2_32f_C3R_Ctx (C function), 1417	nppiNormDiff_L1_8u_AC4R_Ctx (C function), 1431
nppiNorm_L2_32f_C4R_Ctx (C function), 1417	nppiNormDiff_L1_8u_C1MR_Ctx (C function), 1432
nppiNormDiff_Inf_8s_C1MR_Ctx (C function), 1425	nppiNormDiff_L1_8u_C1R_Ctx (C function), 1430
nppiNormDiff_Inf_8s_C3CMR_Ctx (C function), 1425	nppiNormDiff_L1_8u_C3CMR_Ctx (C function), 1432
nppiNormDiff_Inf_8u_AC4R_Ctx (C function), 1423	nppiNormDiff_L1_8u_C3R_Ctx (C function), 1430
nppiNormDiff_Inf_8u_C1MR_Ctx (C function), 1425	nppiNormDiff_L1_8u_C4R_Ctx (C function), 1431
nppiNormDiff_Inf_8u_C1R_Ctx (C function), 1422	nppiNormDiff_L1_16s_AC4R_Ctx (C function), 1431
nppiNormDiff_Inf_8u_C3CMR_Ctx (C function), 1425	nppiNormDiff_L1_16s_C1R_Ctx (C function), 1430
nppiNormDiff_Inf_8u_C3R_Ctx (C function), 1423	nppiNormDiff_L1_16s_C3R_Ctx (C function), 1430
nppiNormDiff_Inf_8u_C4R_Ctx (C function), 1424	nppiNormDiff_L1_16s_C4R_Ctx (C function), 1432
nppiNormDiff_Inf_16s_AC4R_Ctx (C function), 1424	nppiNormDiff_L1_16u_AC4R_Ctx (C function), 1431
nppiNormDiff_Inf_16s_C1R_Ctx (C function), 1422	nppiNormDiff_L1_16u_C1MR_Ctx (C function), 1432
nppiNormDiff_Inf_16s_C3R_Ctx (C function), 1423	nppiNormDiff_L1_16u_C1R_Ctx (C function), 1430
nppiNormDiff_Inf_16s_C4R_Ctx (C function), 1424	nppiNormDiff_L1_16u_C3CMR_Ctx (C function), 1433
nppiNormDiff_Inf_16u_AC4R_Ctx (C function), 1424	nppiNormDiff_L1_16u_C3R_Ctx (C function), 1430
nppiNormDiff_Inf_16u_C1MR_Ctx (C function), 1425	nppiNormDiff_L1_16u_C4R_Ctx (C function), 1431
nppiNormDiff_Inf_16u_C1R_Ctx (C function), 1422	nppiNormDiff_L1_32f_AC4R_Ctx (C function), 1431
nppiNormDiff_Inf_16u_C3CMR_Ctx (C function), 1425	nppiNormDiff_L1_32f_C1MR_Ctx (C function), 1432
nppiNormDiff_Inf_16u_C3R_Ctx (C function), 1423	nppiNormDiff_L1_32f_C1R_Ctx (C function), 1430
nppiNormDiff_Inf_16u_C4R_Ctx (C function), 1424	nppiNormDiff_L1_32f_C3CMR_Ctx (C function), 1433
nppiNormDiff_Inf_32f_AC4R_Ctx (C function), 1424	nppiNormDiff_L1_32f_C3R_Ctx (C function), 1431
nppiNormDiff_Inf_32f_C1MR_Ctx (C function), 1425	nppiNormDiff_L1_32f_C4R_Ctx (C function), 1432
nppiNormDiff_Inf_32f_C1R_Ctx (C function), 1422	nppiNormDiff_L2_8s_C1MR_Ctx (C function), 1439
nppiNormDiff_Inf_32f_C3CMR_Ctx (C function), 1426	nppiNormDiff_L2_8s_C3CMR_Ctx (C function), 1440
nppiNormDiff_Inf_32f_C3R_Ctx (C function), 1423	nppiNormDiff_L2_8u_AC4R_Ctx (C function), 1438
nppiNormDiff_Inf_32f_C4R_Ctx (C function), 1424	nppiNormDiff_L2_8u_C1MR_Ctx (C function), 1439
nppiNormDiff_L1_8s_C1MR_Ctx (C function), 1432	

nppiNormDiff_L2_8u_C1R_Ctx (C function), 1437
 nppiNormDiff_L2_8u_C3CMR_Ctx (C function), 1440
 nppiNormDiff_L2_8u_C3R_Ctx (C function), 1438
 nppiNormDiff_L2_8u_C4R_Ctx (C function), 1439
 nppiNormDiff_L2_16s_AC4R_Ctx (C function), 1438
 nppiNormDiff_L2_16s_C1R_Ctx (C function), 1437
 nppiNormDiff_L2_16s_C3R_Ctx (C function), 1438
 nppiNormDiff_L2_16s_C4R_Ctx (C function), 1439
 nppiNormDiff_L2_16u_AC4R_Ctx (C function), 1438
 nppiNormDiff_L2_16u_C1MR_Ctx (C function), 1439
 nppiNormDiff_L2_16u_C1R_Ctx (C function), 1437
 nppiNormDiff_L2_16u_C3CMR_Ctx (C function), 1440
 nppiNormDiff_L2_16u_C3R_Ctx (C function), 1438
 nppiNormDiff_L2_16u_C4R_Ctx (C function), 1439
 nppiNormDiff_L2_32f_AC4R_Ctx (C function), 1438
 nppiNormDiff_L2_32f_C1MR_Ctx (C function), 1440
 nppiNormDiff_L2_32f_C1R_Ctx (C function), 1437
 nppiNormDiff_L2_32f_C3CMR_Ctx (C function), 1440
 nppiNormDiff_L2_32f_C3R_Ctx (C function), 1438
 nppiNormDiff_L2_32f_C4R_Ctx (C function), 1439
 nppiNormDiffInfGetBufferHostSize_8s_C1MR_Ctx (C function), 1427
 nppiNormDiffInfGetBufferHostSize_8s_C3CMR_Ctx (C function), 1429
 nppiNormDiffInfGetBufferHostSize_8u_AC4R_Ctx (C function), 1428
 nppiNormDiffInfGetBufferHostSize_8u_C1MR_Ctx (C function), 1427
 nppiNormDiffInfGetBufferHostSize_8u_C1R_Ctx (C function), 1426
 nppiNormDiffInfGetBufferHostSize_8u_C3CMR_Ctx (C function), 1429
 nppiNormDiffInfGetBufferHostSize_8u_C3R_Ctx (C function), 1427
 nppiNormDiffInfGetBufferHostSize_8u_C4R_Ctx (C function), 1428
 nppiNormDiffInfGetBufferHostSize_16s_AC4R_Ctx (C function), 1429
 nppiNormDiffInfGetBufferHostSize_16s_C1R_Ctx (C function), 1426
 nppiNormDiffInfGetBufferHostSize_16s_C3R_Ctx (C function), 1428
 nppiNormDiffInfGetBufferHostSize_16s_C4R_Ctx (C function), 1428
 nppiNormDiffInfGetBufferHostSize_16u_AC4R_Ctx (C function), 1429
 nppiNormDiffInfGetBufferHostSize_16u_C1MR_Ctx (C function), 1427
 nppiNormDiffInfGetBufferHostSize_16u_C1R_Ctx (C function), 1426
 nppiNormDiffInfGetBufferHostSize_16u_C3CMR_Ctx (C function), 1429
 nppiNormDiffInfGetBufferHostSize_16u_C3R_Ctx (C function), 1427
 nppiNormDiffInfGetBufferHostSize_16u_C4R_Ctx (C function), 1428
 nppiNormDiffInfGetBufferHostSize_32f_AC4R_Ctx (C function), 1429
 nppiNormDiffInfGetBufferHostSize_32f_C1MR_Ctx (C function), 1427
 nppiNormDiffInfGetBufferHostSize_32f_C1R_Ctx (C function), 1427
 nppiNormDiffInfGetBufferHostSize_32f_C3CMR_Ctx (C function), 1429
 nppiNormDiffInfGetBufferHostSize_32f_C3R_Ctx (C function), 1428
 nppiNormDiffInfGetBufferHostSize_32f_C4R_Ctx (C function), 1428
 nppiNormDiffL1GetBufferHostSize_8s_C1MR_Ctx (C function), 1434
 nppiNormDiffL1GetBufferHostSize_8s_C3CMR_Ctx (C function), 1436
 nppiNormDiffL1GetBufferHostSize_8u_AC4R_Ctx (C function), 1436
 nppiNormDiffL1GetBufferHostSize_8u_C1MR_Ctx (C function), 1434
 nppiNormDiffL1GetBufferHostSize_8u_C1R_Ctx (C function), 1434
 nppiNormDiffL1GetBufferHostSize_8u_C3CMR_Ctx (C function), 1436
 nppiNormDiffL1GetBufferHostSize_8u_C3R_Ctx (C function), 1435
 nppiNormDiffL1GetBufferHostSize_8u_C4R_Ctx (C function), 1435
 nppiNormDiffL1GetBufferHostSize_16s_AC4R_Ctx (C function), 1436
 nppiNormDiffL1GetBufferHostSize_16s_C1R_Ctx (C function), 1434

nppiNormDiffL1GetBufferHostSize_16s_C3R_0pp (C function), 1435
 nppiNormDiffL1GetBufferHostSize_16s_C4R_0pp (C function), 1435
 nppiNormDiffL1GetBufferHostSize_16u_AC4R_0pp (C function), 1436
 nppiNormDiffL1GetBufferHostSize_16u_C1MR_0pp (C function), 1434
 nppiNormDiffL1GetBufferHostSize_16u_C1R_0pp (C function), 1434
 nppiNormDiffL1GetBufferHostSize_16u_C3CMR_0pp (C function), 1437
 nppiNormDiffL1GetBufferHostSize_16u_C3R_0pp (C function), 1435
 nppiNormDiffL1GetBufferHostSize_16u_C4R_0pp (C function), 1435
 nppiNormDiffL1GetBufferHostSize_32f_AC4R_0pp (C function), 1436
 nppiNormDiffL1GetBufferHostSize_32f_C1MR_0pp (C function), 1434
 nppiNormDiffL1GetBufferHostSize_32f_C1R_0pp (C function), 1434
 nppiNormDiffL1GetBufferHostSize_32f_C3CMR_0pp (C function), 1437
 nppiNormDiffL1GetBufferHostSize_32f_C3R_0pp (C function), 1435
 nppiNormDiffL1GetBufferHostSize_32f_C4R_0pp (C function), 1436
 nppiNormDiffL2GetBufferHostSize_8s_C1MR_0pp (C function), 1442
 nppiNormDiffL2GetBufferHostSize_8s_C3CMR_0pp (C function), 1444
 nppiNormDiffL2GetBufferHostSize_8u_AC4R_0pp (C function), 1443
 nppiNormDiffL2GetBufferHostSize_8u_C1MR_0pp (C function), 1441
 nppiNormDiffL2GetBufferHostSize_8u_C1R_0pp (C function), 1441
 nppiNormDiffL2GetBufferHostSize_8u_C3CMR_0pp (C function), 1444
 nppiNormDiffL2GetBufferHostSize_8u_C3R_0pp (C function), 1442
 nppiNormDiffL2GetBufferHostSize_8u_C4R_0pp (C function), 1442
 nppiNormDiffL2GetBufferHostSize_16s_AC4R_0pp (C function), 1443
 nppiNormDiffL2GetBufferHostSize_16s_C1R_0pp (C function), 1441
 nppiNormDiffL2GetBufferHostSize_16s_C3R_0pp (C function), 1442
 nppiNormDiffL2GetBufferHostSize_16s_C4R_0pp (C function), 1443
 nppiNormDiffL2GetBufferHostSize_16u_AC4R_0pp (C function), 1443
 nppiNormDiffL2GetBufferHostSize_16u_C1MR_0pp (C function), 1442
 nppiNormDiffL2GetBufferHostSize_16u_C1R_0pp (C function), 1443
 nppiNormDiffL2GetBufferHostSize_16u_C3CMR_0pp (C function), 1443
 nppiNormDiffL2GetBufferHostSize_16u_C3R_0pp (C function), 1442
 nppiNormDiffL2GetBufferHostSize_16u_C4R_0pp (C function), 1443
 nppiNormDiffL2GetBufferHostSize_32f_AC4R_0pp (C function), 1443
 nppiNormDiffL2GetBufferHostSize_32f_C1MR_0pp (C function), 1442
 nppiNormDiffL2GetBufferHostSize_32f_C1R_0pp (C function), 1441
 nppiNormDiffL2GetBufferHostSize_32f_C3CMR_0pp (C function), 1444
 nppiNormDiffL2GetBufferHostSize_32f_C3R_0pp (C function), 1442
 nppiNormDiffL2GetBufferHostSize_32f_C4R_0pp (C function), 1443
 nppiNormInfGetBufferHostSize_8s_C1MR_0pp (C function), 1407
 nppiNormInfGetBufferHostSize_8s_C3CMR_0pp (C function), 1409
 nppiNormInfGetBufferHostSize_8u_AC4R_0pp (C function), 1408
 nppiNormInfGetBufferHostSize_8u_C1MR_0pp (C function), 1407
 nppiNormInfGetBufferHostSize_8u_C1R_0pp (C function), 1406
 nppiNormInfGetBufferHostSize_8u_C3CMR_0pp (C function), 1408
 nppiNormInfGetBufferHostSize_8u_C3R_0pp (C function), 1407
 nppiNormInfGetBufferHostSize_8u_C4R_0pp (C function), 1408
 nppiNormInfGetBufferHostSize_16s_AC4R_0pp (C function), 1408
 nppiNormInfGetBufferHostSize_16s_C1R_0pp (C function), 1406
 nppiNormInfGetBufferHostSize_16s_C3R_0pp (C function), 1407
 nppiNormInfGetBufferHostSize_16s_C4R_0pp (C function), 1408
 nppiNormInfGetBufferHostSize_16u_AC4R_0pp (C function), 1408
 nppiNormInfGetBufferHostSize_16u_C1MR_0pp (C function), 1407
 nppiNormInfGetBufferHostSize_16u_C1R_0pp (C function), 1406
 nppiNormInfGetBufferHostSize_16u_C3CMR_0pp (C function), 1409

nppiNormInfGetBufferHostSize_16u_C3R_Ctx nppiNormL1GetBufferHostSize_32f_AC4R_Ctx
 (C function), 1407 (C function), 1414
 nppiNormInfGetBufferHostSize_16u_C4R_Ctx nppiNormL1GetBufferHostSize_32f_C1MR_Ctx
 (C function), 1408 (C function), 1413
 nppiNormInfGetBufferHostSize_32f_AC4R_Ctx nppiNormL1GetBufferHostSize_32f_C1R_Ctx
 (C function), 1408 (C function), 1413
 nppiNormInfGetBufferHostSize_32f_C1MR_Ctx nppiNormL1GetBufferHostSize_32f_C3CMR_Ctx
 (C function), 1407 (C function), 1415
 nppiNormInfGetBufferHostSize_32f_C1R_Ctx nppiNormL1GetBufferHostSize_32f_C3R_Ctx
 (C function), 1407 (C function), 1414
 nppiNormInfGetBufferHostSize_32f_C3CMR_Ctx nppiNormL1GetBufferHostSize_32f_C4R_Ctx
 (C function), 1409 (C function), 1415
 nppiNormInfGetBufferHostSize_32f_C3R_Ctx nppiNormL2GetBufferHostSize_8s_C1MR_Ctx
 (C function), 1407 (C function), 1420
 nppiNormInfGetBufferHostSize_32f_C4R_Ctx nppiNormL2GetBufferHostSize_8s_C3CMR_Ctx
 (C function), 1408 (C function), 1421
 nppiNormInfGetBufferHostSize_32s_C1R_Ctx nppiNormL2GetBufferHostSize_8u_AC4R_Ctx
 (C function), 1406 (C function), 1420
 nppiNormL1GetBufferHostSize_8s_C1MR_Ctx nppiNormL2GetBufferHostSize_8u_C1MR_Ctx
 (C function), 1413 (C function), 1419
 nppiNormL1GetBufferHostSize_8s_C3CMR_Ctx nppiNormL2GetBufferHostSize_8u_C1R_Ctx
 (C function), 1415 (C function), 1419
 nppiNormL1GetBufferHostSize_8u_AC4R_Ctx nppiNormL2GetBufferHostSize_8u_C3CMR_Ctx
 (C function), 1414 (C function), 1421
 nppiNormL1GetBufferHostSize_8u_C1MR_Ctx nppiNormL2GetBufferHostSize_8u_C3R_Ctx
 (C function), 1413 (C function), 1420
 nppiNormL1GetBufferHostSize_8u_C1R_Ctx nppiNormL2GetBufferHostSize_8u_C4R_Ctx
 (C function), 1413 (C function), 1421
 nppiNormL1GetBufferHostSize_8u_C3CMR_Ctx nppiNormL2GetBufferHostSize_16s_AC4R_Ctx
 (C function), 1415 (C function), 1421
 nppiNormL1GetBufferHostSize_8u_C3R_Ctx nppiNormL2GetBufferHostSize_16s_C1R_Ctx
 (C function), 1413 (C function), 1419
 nppiNormL1GetBufferHostSize_8u_C4R_Ctx nppiNormL2GetBufferHostSize_16s_C3R_Ctx
 (C function), 1414 (C function), 1420
 nppiNormL1GetBufferHostSize_16s_AC4R_Ctx nppiNormL2GetBufferHostSize_16s_C4R_Ctx
 (C function), 1414 (C function), 1421
 nppiNormL1GetBufferHostSize_16s_C1R_Ctx nppiNormL2GetBufferHostSize_16u_AC4R_Ctx
 (C function), 1413 (C function), 1421
 nppiNormL1GetBufferHostSize_16s_C3R_Ctx nppiNormL2GetBufferHostSize_16u_C1MR_Ctx
 (C function), 1414 (C function), 1420
 nppiNormL1GetBufferHostSize_16s_C4R_Ctx nppiNormL2GetBufferHostSize_16u_C1R_Ctx
 (C function), 1415 (C function), 1419
 nppiNormL1GetBufferHostSize_16u_AC4R_Ctx nppiNormL2GetBufferHostSize_16u_C3CMR_Ctx
 (C function), 1414 (C function), 1422
 nppiNormL1GetBufferHostSize_16u_C1MR_Ctx nppiNormL2GetBufferHostSize_16u_C3R_Ctx
 (C function), 1413 (C function), 1420
 nppiNormL1GetBufferHostSize_16u_C1R_Ctx nppiNormL2GetBufferHostSize_16u_C4R_Ctx
 (C function), 1413 (C function), 1421
 nppiNormL1GetBufferHostSize_16u_C3CMR_Ctx nppiNormL2GetBufferHostSize_32f_AC4R_Ctx
 (C function), 1415 (C function), 1421
 nppiNormL1GetBufferHostSize_16u_C3R_Ctx nppiNormL2GetBufferHostSize_32f_C1MR_Ctx
 (C function), 1414 (C function), 1420
 nppiNormL1GetBufferHostSize_16u_C4R_Ctx nppiNormL2GetBufferHostSize_32f_C1R_Ctx
 (C function), 1415 (C function), 1419

nppiNormL2GetBufferHostSize_32f_C3CMR_Ctx	1445	nppiNormRel_Inf_32f_C4R_Ctx	(C function),
(C function), 1422			
nppiNormL2GetBufferHostSize_32f_C3R_Ctx	1446	nppiNormRel_L1_8s_C1MR_Ctx	(C function),
(C function), 1420			
nppiNormL2GetBufferHostSize_32f_C4R_Ctx	1454	nppiNormRel_L1_8s_C3CMR_Ctx	(C function),
(C function), 1421			
NppiNorm.nppiNormInf (C enumerator), 41	1455	nppiNormRel_L1_8u_AC4R_Ctx	(C function),
NppiNorm.nppiNormL1 (C enumerator), 41			
NppiNorm.nppiNormL2 (C enumerator), 41	1453	nppiNormRel_L1_8u_C1MR_Ctx	(C function),
nppiNormRel_Inf_8s_C1MR_Ctx (C function),	1454		
1447		nppiNormRel_L1_8u_C1R_Ctx	(C function),
nppiNormRel_Inf_8s_C3CMR_Ctx (C function),	1452		
1447		nppiNormRel_L1_8u_C3CMR_Ctx	(C function),
nppiNormRel_Inf_8u_AC4R_Ctx (C function),	1454		
1445		nppiNormRel_L1_8u_C3R_Ctx	(C function),
nppiNormRel_Inf_8u_C1MR_Ctx (C function),	1452		
1447		nppiNormRel_L1_8u_C4R_Ctx	(C function),
nppiNormRel_Inf_8u_C1R_Ctx (C function),	1453		
1444		nppiNormRel_L1_16s_AC4R_Ctx	(C function),
nppiNormRel_Inf_8u_C3CMR_Ctx (C function),	1453		
1447		nppiNormRel_L1_16s_C1R_Ctx	(C function),
nppiNormRel_Inf_8u_C3R_Ctx (C function),	1452		
1445		nppiNormRel_L1_16s_C3R_Ctx	(C function),
nppiNormRel_Inf_8u_C4R_Ctx (C function),	1452		
1446		nppiNormRel_L1_16s_C4R_Ctx	(C function),
nppiNormRel_Inf_16s_AC4R_Ctx (C function),	1454		
1446		nppiNormRel_L1_16u_AC4R_Ctx	(C function),
nppiNormRel_Inf_16s_C1R_Ctx (C function),	1453		
1445		nppiNormRel_L1_16u_C1MR_Ctx	(C function),
nppiNormRel_Inf_16s_C3R_Ctx (C function),	1454		
1445		nppiNormRel_L1_16u_C1R_Ctx	(C function),
nppiNormRel_Inf_16s_C4R_Ctx (C function),	1452		
1446		nppiNormRel_L1_16u_C3CMR_Ctx	(C function),
nppiNormRel_Inf_16u_AC4R_Ctx (C function),	1455		
1446		nppiNormRel_L1_16u_C3R_Ctx	(C function),
nppiNormRel_Inf_16u_C1MR_Ctx (C function),	1452		
1447		nppiNormRel_L1_16u_C4R_Ctx	(C function),
nppiNormRel_Inf_16u_C1R_Ctx (C function),	1453		
1444		nppiNormRel_L1_32f_AC4R_Ctx	(C function),
nppiNormRel_Inf_16u_C3CMR_Ctx (C function),	1453		
1447		nppiNormRel_L1_32f_C1MR_Ctx	(C function),
nppiNormRel_Inf_16u_C3R_Ctx (C function),	1454		
1445		nppiNormRel_L1_32f_C1R_Ctx	(C function),
nppiNormRel_Inf_16u_C4R_Ctx (C function),	1452		
1446		nppiNormRel_L1_32f_C3CMR_Ctx	(C function),
nppiNormRel_Inf_32f_AC4R_Ctx (C function),	1455		
1446		nppiNormRel_L1_32f_C3R_Ctx	(C function),
nppiNormRel_Inf_32f_C1MR_Ctx (C function),	1453		
1447		nppiNormRel_L1_32f_C4R_Ctx	(C function),
nppiNormRel_Inf_32f_C1R_Ctx (C function),	1454		
1445		nppiNormRel_L2_8s_C1MR_Ctx	(C function),
nppiNormRel_Inf_32f_C3CMR_Ctx (C function),	1461		
1448		nppiNormRel_L2_8s_C3CMR_Ctx	(C function),
nppiNormRel_Inf_32f_C3R_Ctx (C function),			

1462
 nppiNormRel_L2_8u_AC4R_Ctx (C function), nppiNormRelInfGetBufferHostSize_8u_C3CMR_Ctx
 1460 (C function), 1451
 nppiNormRel_L2_8u_C1MR_Ctx (C function), nppiNormRelInfGetBufferHostSize_8u_C3R_Ctx
 1461 (C function), 1449
 nppiNormRel_L2_8u_C1R_Ctx (C function), nppiNormRelInfGetBufferHostSize_8u_C4R_Ctx
 1459 (C function), 1450
 nppiNormRel_L2_8u_C3CMR_Ctx (C function), nppiNormRelInfGetBufferHostSize_16s_AC4R_Ctx
 1461 (C function), 1451
 nppiNormRel_L2_8u_C3R_Ctx (C function), nppiNormRelInfGetBufferHostSize_16s_C1R_Ctx
 1459 (C function), 1448
 nppiNormRel_L2_8u_C4R_Ctx (C function), nppiNormRelInfGetBufferHostSize_16s_C3R_Ctx
 1460 (C function), 1449
 nppiNormRel_L2_16s_AC4R_Ctx (C function), nppiNormRelInfGetBufferHostSize_16s_C4R_Ctx
 1460 (C function), 1450
 nppiNormRel_L2_16s_C1R_Ctx (C function), nppiNormRelInfGetBufferHostSize_16u_AC4R_Ctx
 1459 (C function), 1450
 nppiNormRel_L2_16s_C3R_Ctx (C function), nppiNormRelInfGetBufferHostSize_16u_C1MR_Ctx
 1459 (C function), 1449
 nppiNormRel_L2_16s_C4R_Ctx (C function), nppiNormRelInfGetBufferHostSize_16u_C1R_Ctx
 1461 (C function), 1448
 nppiNormRel_L2_16u_AC4R_Ctx (C function), nppiNormRelInfGetBufferHostSize_16u_C3CMR_Ctx
 1460 (C function), 1451
 nppiNormRel_L2_16u_C1MR_Ctx (C function), nppiNormRelInfGetBufferHostSize_16u_C3R_Ctx
 1461 (C function), 1449
 nppiNormRel_L2_16u_C1R_Ctx (C function), nppiNormRelInfGetBufferHostSize_16u_C4R_Ctx
 1459 (C function), 1450
 nppiNormRel_L2_16u_C3CMR_Ctx (C function), nppiNormRelInfGetBufferHostSize_32f_AC4R_Ctx
 1462 (C function), 1451
 nppiNormRel_L2_16u_C3R_Ctx (C function), nppiNormRelInfGetBufferHostSize_32f_C1MR_Ctx
 1459 (C function), 1449
 nppiNormRel_L2_16u_C4R_Ctx (C function), nppiNormRelInfGetBufferHostSize_32f_C1R_Ctx
 1460 (C function), 1449
 nppiNormRel_L2_32f_AC4R_Ctx (C function), nppiNormRelInfGetBufferHostSize_32f_C3CMR_Ctx
 1460 (C function), 1451
 nppiNormRel_L2_32f_C1MR_Ctx (C function), nppiNormRelInfGetBufferHostSize_32f_C3R_Ctx
 1461 (C function), 1450
 nppiNormRel_L2_32f_C1R_Ctx (C function), nppiNormRelInfGetBufferHostSize_32f_C4R_Ctx
 1459 (C function), 1450
 nppiNormRel_L2_32f_C3CMR_Ctx (C function), nppiNormRelL1GetBufferHostSize_8s_C1MR_Ctx
 1462 (C function), 1456
 nppiNormRel_L2_32f_C3R_Ctx (C function), nppiNormRelL1GetBufferHostSize_8s_C3CMR_Ctx
 1460 (C function), 1458
 nppiNormRel_L2_32f_C4R_Ctx (C function), nppiNormRelL1GetBufferHostSize_8u_AC4R_Ctx
 1461 (C function), 1457
 nppiNormRelInfGetBufferHostSize_8s_C1MR_Ctx nppiNormRelL1GetBufferHostSize_8u_C1MR_Ctx
 (C function), 1449 (C function), 1456
 nppiNormRelInfGetBufferHostSize_8s_C3CMR_Ctx nppiNormRelL1GetBufferHostSize_8u_C1R_Ctx
 (C function), 1451 (C function), 1456
 nppiNormRelInfGetBufferHostSize_8u_AC4R_Ctx nppiNormRelL1GetBufferHostSize_8u_C3CMR_Ctx
 (C function), 1450 (C function), 1458
 nppiNormRelInfGetBufferHostSize_8u_C1MR_Ctx nppiNormRelL1GetBufferHostSize_8u_C3R_Ctx
 (C function), 1449 (C function), 1457
 nppiNormRelInfGetBufferHostSize_8u_C1R_Ctx nppiNormRelL1GetBufferHostSize_8u_C4R_Ctx

<code>nppiNormRelL1GetBufferHostSize_16s_AC4R_0p</code> (C function), 1457	<code>nppiNormRelL2GetBufferHostSize_16s_C4R_Ctx</code> (C function), 1464
<code>nppiNormRelL1GetBufferHostSize_16s_C1R_Ctx</code> (C function), 1456	<code>nppiNormRelL2GetBufferHostSize_16u_AC4R_Ctx</code> (C function), 1465
<code>nppiNormRelL1GetBufferHostSize_16s_C3R_Ctx</code> (C function), 1457	<code>nppiNormRelL2GetBufferHostSize_16u_C1MR_Ctx</code> (C function), 1463
<code>nppiNormRelL1GetBufferHostSize_16s_C4R_Ctx</code> (C function), 1457	<code>nppiNormRelL2GetBufferHostSize_16u_C1R_Ctx</code> (C function), 1463
<code>nppiNormRelL1GetBufferHostSize_16u_AC4R_0p</code> (C function), 1458	<code>nppiNormRelL2GetBufferHostSize_16u_C3CMR_Ctx</code> (C function), 1465
<code>nppiNormRelL1GetBufferHostSize_16u_C1MR_0p</code> (C function), 1456	<code>nppiNormRelL2GetBufferHostSize_16u_C3R_Ctx</code> (C function), 1464
<code>nppiNormRelL1GetBufferHostSize_16u_C1R_Ctx</code> (C function), 1456	<code>nppiNormRelL2GetBufferHostSize_16u_C4R_Ctx</code> (C function), 1464
<code>nppiNormRelL1GetBufferHostSize_16u_C3CMR_0p</code> (C function), 1458	<code>nppiNormRelL2GetBufferHostSize_32f_AC4R_Ctx</code> (C function), 1465
<code>nppiNormRelL1GetBufferHostSize_16u_C3R_Ctx</code> (C function), 1457	<code>nppiNormRelL2GetBufferHostSize_32f_C1MR_Ctx</code> (C function), 1463
<code>nppiNormRelL1GetBufferHostSize_16u_C4R_Ctx</code> (C function), 1457	<code>nppiNormRelL2GetBufferHostSize_32f_C1R_Ctx</code> (C function), 1463
<code>nppiNormRelL1GetBufferHostSize_32f_AC4R_0p</code> (C function), 1458	<code>nppiNormRelL2GetBufferHostSize_32f_C3CMR_Ctx</code> (C function), 1465
<code>nppiNormRelL1GetBufferHostSize_32f_C1MR_0p</code> (C function), 1456	<code>nppiNormRelL2GetBufferHostSize_32f_C3R_Ctx</code> (C function), 1464
<code>nppiNormRelL1GetBufferHostSize_32f_C1R_Ctx</code> (C function), 1456	<code>nppiNormRelL2GetBufferHostSize_32f_C4R_Ctx</code> (C function), 1464
<code>nppiNormRelL1GetBufferHostSize_32f_C3CMR_0p</code> (C function), 1458	<code>nppiNot_8u_AC4IR_Ctx</code> (C function), 490
<code>nppiNormRelL1GetBufferHostSize_32f_C3R_Ctx</code> (C function), 1457	<code>nppiNot_8u_AC4R_Ctx</code> (C function), 490
<code>nppiNormRelL1GetBufferHostSize_32f_C4R_Ctx</code> (C function), 1457	<code>nppiNot_8u_C1IR_Ctx</code> (C function), 489
<code>nppiNormRelL2GetBufferHostSize_8s_C1MR_Ctx</code> (C function), 1463	<code>nppiNot_8u_C1R_Ctx</code> (C function), 489
<code>nppiNormRelL2GetBufferHostSize_8s_C3CMR_0p</code> (C function), 1465	<code>nppiNot_8u_C3IR_Ctx</code> (C function), 490
<code>nppiNormRelL2GetBufferHostSize_8u_AC4R_Ctx</code> (C function), 1464	<code>nppiNot_8u_C3R_Ctx</code> (C function), 489
<code>nppiNormRelL2GetBufferHostSize_8u_C1MR_Ctx</code> (C function), 1463	<code>nppiNot_8u_C4IR_Ctx</code> (C function), 491
<code>nppiNormRelL2GetBufferHostSize_8u_C1R_Ctx</code> (C function), 1463	<code>nppiNot_8u_C4R_Ctx</code> (C function), 491
<code>nppiNormRelL2GetBufferHostSize_8u_C3CMR_0p</code> (C function), 1465	<code>nppiNV12_8u_ToUYVP_10u_P2C3R_Ctx</code> (C function), 531
<code>nppiNormRelL2GetBufferHostSize_8u_C3R_Ctx</code> (C function), 1464	<code>nppiNV12ToBGR_8u_P2C3R_Ctx</code> (C function), 723
<code>nppiNormRelL2GetBufferHostSize_8u_C4R_Ctx</code> (C function), 1464	<code>nppiNV12ToBGR_8u_P2C4R_Ctx</code> (C function), 534
<code>nppiNormRelL2GetBufferHostSize_8u_C4CMR_0p</code> (C function), 1465	<code>nppiNV12ToBGR_601_8u_P2C3R_Ctx</code> (C function), 724
<code>nppiNormRelL2GetBufferHostSize_8u_C4R_Ctx</code> (C function), 1464	<code>nppiNV12ToBGR_709CSC_8u_P2C3R_Ctx</code> (C function), 724
<code>nppiNormRelL2GetBufferHostSize_8u_C4CMR_0p</code> (C function), 1465	<code>nppiNV12ToBGR_709HDTV_8u_P2C3R_Ctx</code> (C function), 723
<code>nppiNormRelL2GetBufferHostSize_8u_C4R_Ctx</code> (C function), 1464	<code>nppiNV12ToBGR_BT2020_8u_P2C3R_Ctx</code> (C function), 534
<code>nppiNormRelL2GetBufferHostSize_16s_AC4R_0p</code> (C function), 1465	<code>nppiNV12ToRGB_8u_ColorTwist32f_P2C3R_Ctx</code> (C function), 527
<code>nppiNormRelL2GetBufferHostSize_16s_C1R_Ctx</code> (C function), 1463	<code>nppiNV12ToRGB_8u_P2C3R_Ctx</code> (C function), 718
<code>nppiNormRelL2GetBufferHostSize_16s_C3R_Ctx</code> (C function), 1463	<code>nppiNV12ToRGB_8u_P2C4R_Ctx</code> (C function),

534
 nppiNV12ToRGB_16u_ColorTwist32f_P2C3R_Ctx (C function), 527
 nppiNV12ToRGB_601_8u_P2C3R_Ctx (C function), 719
 nppiNV12ToRGB_709CSC_8u_P2C3R_Ctx (C function), 718
 nppiNV12ToRGB_709HDTV_8u_P2C3R_Ctx (C function), 718
 nppiNV12ToRGB_BT2020_8u_P2C3R_Ctx (C function), 534
 nppiNV12ToYUV420_8u_P2P3R_Ctx (C function), 693
 nppiNV16_8u_ToUYVP_10u_P2C3R_Ctx (C function), 532
 nppiNV21ToBGR_8u_P2C4R_Ctx (C function), 725
 nppiNV21ToRGB_8u_P2C4R_Ctx (C function), 720
 nppiOr_8u_AC4IR_Ctx (C function), 472
 nppiOr_8u_AC4R_Ctx (C function), 472
 nppiOr_8u_C1IR_Ctx (C function), 471
 nppiOr_8u_C1R_Ctx (C function), 470
 nppiOr_8u_C3IR_Ctx (C function), 471
 nppiOr_8u_C3R_Ctx (C function), 471
 nppiOr_8u_C4IR_Ctx (C function), 473
 nppiOr_8u_C4R_Ctx (C function), 472
 nppiOr_16u_AC4IR_Ctx (C function), 475
 nppiOr_16u_AC4R_Ctx (C function), 475
 nppiOr_16u_C1IR_Ctx (C function), 474
 nppiOr_16u_C1R_Ctx (C function), 473
 nppiOr_16u_C3IR_Ctx (C function), 474
 nppiOr_16u_C3R_Ctx (C function), 474
 nppiOr_16u_C4IR_Ctx (C function), 476
 nppiOr_16u_C4R_Ctx (C function), 475
 nppiOr_32s_AC4IR_Ctx (C function), 478
 nppiOr_32s_AC4R_Ctx (C function), 478
 nppiOr_32s_C1IR_Ctx (C function), 477
 nppiOr_32s_C1R_Ctx (C function), 476
 nppiOr_32s_C3IR_Ctx (C function), 477
 nppiOr_32s_C3R_Ctx (C function), 477
 nppiOr_32s_C4IR_Ctx (C function), 479
 nppiOr_32s_C4R_Ctx (C function), 478
 nppiOrC_8u_AC4IR_Ctx (C function), 423
 nppiOrC_8u_AC4R_Ctx (C function), 422
 nppiOrC_8u_C1IR_Ctx (C function), 421
 nppiOrC_8u_C1R_Ctx (C function), 421
 nppiOrC_8u_C3IR_Ctx (C function), 422
 nppiOrC_8u_C3R_Ctx (C function), 422
 nppiOrC_8u_C4IR_Ctx (C function), 424
 nppiOrC_8u_C4R_Ctx (C function), 423
 nppiOrC_16u_AC4IR_Ctx (C function), 426
 nppiOrC_16u_AC4R_Ctx (C function), 425
 nppiOrC_16u_C1IR_Ctx (C function), 424
 nppiOrC_16u_C1R_Ctx (C function), 424
 nppiOrC_16u_C3IR_Ctx (C function), 425
 nppiOrC_16u_C3R_Ctx (C function), 425
 nppiOrC_16u_C4IR_Ctx (C function), 426
 nppiOrC_16u_C4R_Ctx (C function), 426
 nppiOrC_32s_AC4IR_Ctx (C function), 428
 nppiOrC_32s_AC4R_Ctx (C function), 428
 nppiOrC_32s_C1IR_Ctx (C function), 427
 nppiOrC_32s_C1R_Ctx (C function), 427
 nppiOrC_32s_C3IR_Ctx (C function), 428
 nppiOrC_32s_C3R_Ctx (C function), 427
 nppiOrC_32s_C4IR_Ctx (C function), 429
 nppiOrC_32s_C4R_Ctx (C function), 429
 nppiP010_16u_ToUYVP_10u_P3C3R_Ctx (C function), 531
 nppiP010ToRGB_8u_P2C3R_Ctx (C function), 535
 nppiP010ToRGB_8u_P2C4R_Ctx (C function), 535
 nppiP010ToRGB_16u_P2C3R_Ctx (C function), 535
 nppiP010ToRGB_16u_P2C4R_Ctx (C function), 535
 nppiP016ToRGB_16u_P2C3R_Ctx (C function), 535
 nppiP016ToRGB_16u_P2C4R_Ctx (C function), 535
 nppiP208_8u_ToUYVP_10u_C3C3R_Ctx (C function), 532
 nppiP210_16u_ToUYVP_10u_P2C3R_Ctx (C function), 532
 NppiPoint (C struct), 33
 NppiPoint32f (C struct), 34
 NppiPoint32f.x (C var), 34
 NppiPoint32f.y (C var), 34
 NppiPoint64f (C struct), 34
 NppiPoint64f.x (C var), 34
 NppiPoint64f.y (C var), 34
 NppiPoint.x (C var), 34
 NppiPoint.y (C var), 34
 nppiPSNR_8u_C1R_Ctx (C function), 1641
 nppiPSNR_8u_C3R_Ctx (C function), 1641
 nppiPSNRBatch_8u_C1R_Advanced_Ctx (C function), 1655
 nppiPSNRBatch_8u_C1R_Ctx (C function), 1648
 nppiPSNRBatch_8u_C3R_Advanced_Ctx (C function), 1655
 nppiPSNRBatch_8u_C3R_Ctx (C function), 1648
 nppiPSNRBatchGetBufferHostSize_8u_C1R_Ctx (C function), 1652
 nppiPSNRBatchGetBufferHostSize_8u_C3R_Ctx (C function), 1653
 nppiPSNRGetBufferHostSize_8u_C1R_Ctx (C function), 1645

nppiPSNRGetBufferHostSize_8u_C3R_Ctx (C function), 1645
 nppiQualityIndex_8u32f_AC4R_Ctx (C function), 1577
 nppiQualityIndex_8u32f_C1R_Ctx (C function), 1576
 nppiQualityIndex_8u32f_C3R_Ctx (C function), 1576
 nppiQualityIndex_16u32f_AC4R_Ctx (C function), 1577
 nppiQualityIndex_16u32f_C1R_Ctx (C function), 1576
 nppiQualityIndex_16u32f_C3R_Ctx (C function), 1576
 nppiQualityIndex_32f_AC4R_Ctx (C function), 1577
 nppiQualityIndex_32f_C1R_Ctx (C function), 1576
 nppiQualityIndex_32f_C3R_Ctx (C function), 1576
 nppiQualityIndexGetBufferHostSize_8u32f_AC4R_Ctx (C function), 1578
 nppiQualityIndexGetBufferHostSize_8u32f_C1R_Ctx (C function), 1578
 nppiQualityIndexGetBufferHostSize_8u32f_C3R_Ctx (C function), 1578
 nppiQualityIndexGetBufferHostSize_16u32f_AC4R_Ctx (C function), 1578
 nppiQualityIndexGetBufferHostSize_16u32f_C1R_Ctx (C function), 1578
 nppiQualityIndexGetBufferHostSize_16u32f_C3R_Ctx (C function), 1578
 nppiQualityIndexGetBufferHostSize_32f_AC4R_Ctx (C function), 1579
 nppiQualityIndexGetBufferHostSize_32f_C1R_Ctx (C function), 1578
 nppiQualityIndexGetBufferHostSize_32f_C3R_Ctx (C function), 1578
 NppiRect (C struct), 35
 NppiRect.height (C var), 35
 nppiRectStdDev_32f_C1R_Ctx (C function), 1487
 nppiRectStdDev_32s32f_C1R_Ctx (C function), 1488
 nppiRectStdDev_32s_C1RSfs_Ctx (C function), 1488
 NppiRect.width (C var), 35
 NppiRect.x (C var), 35
 NppiRect.y (C var), 35
 nppiRemap_8u_AC4R_Ctx (C function), 1251
 nppiRemap_8u_C1R_Ctx (C function), 1251
 nppiRemap_8u_C3R_Ctx (C function), 1251
 nppiRemap_8u_C4R_Ctx (C function), 1251
 nppiRemap_8u_P3R_Ctx (C function), 1251
 nppiRemap_8u_P4R_Ctx (C function), 1251
 nppiRemap_16s_AC4R_Ctx (C function), 1253
 nppiRemap_16s_C1R_Ctx (C function), 1253
 nppiRemap_16s_C3R_Ctx (C function), 1253
 nppiRemap_16s_C4R_Ctx (C function), 1253
 nppiRemap_16s_P3R_Ctx (C function), 1253
 nppiRemap_16s_P4R_Ctx (C function), 1254
 nppiRemap_16u_AC4R_Ctx (C function), 1252
 nppiRemap_16u_C1R_Ctx (C function), 1252
 nppiRemap_16u_C3R_Ctx (C function), 1252
 nppiRemap_16u_C4R_Ctx (C function), 1252
 nppiRemap_16u_P3R_Ctx (C function), 1252
 nppiRemap_16u_P4R_Ctx (C function), 1252
 nppiRemap_32f_AC4R_Ctx (C function), 1254
 nppiRemap_32f_C1R_Ctx (C function), 1254
 nppiRemap_32f_C3R_Ctx (C function), 1254
 nppiRemap_32f_C4R_Ctx (C function), 1254
 nppiRemap_32f_P3R_Ctx (C function), 1254
 nppiRemap_32f_P4R_Ctx (C function), 1255
 nppiRemap_64f_AC4R_Ctx (C function), 1255
 nppiRemap_64f_C1R_Ctx (C function), 1255
 nppiRemap_64f_C3R_Ctx (C function), 1255
 nppiRemap_64f_C4R_Ctx (C function), 1255
 nppiRemap_64f_P3R_Ctx (C function), 1255
 nppiRemap_64f_P4R_Ctx (C function), 1256
 nppiResize_8u_AC4R_Ctx (C function), 1237
 nppiResize_8u_C1R_Ctx (C function), 1237
 nppiResize_8u_C3R_Ctx (C function), 1237
 nppiResize_8u_C4R_Ctx (C function), 1237
 nppiResize_8u_P3R_Ctx (C function), 1238
 nppiResize_8u_P4R_Ctx (C function), 1238
 nppiResize_16f_C1R_Ctx (C function), 1240
 nppiResize_16f_C3R_Ctx (C function), 1240
 nppiResize_16f_C4R_Ctx (C function), 1240
 nppiResize_16s_AC4R_Ctx (C function), 1239
 nppiResize_16s_C1R_Ctx (C function), 1239
 nppiResize_16s_C3R_Ctx (C function), 1239
 nppiResize_16s_C4R_Ctx (C function), 1239
 nppiResize_16s_P3R_Ctx (C function), 1239
 nppiResize_16s_P4R_Ctx (C function), 1240
 nppiResize_16u_AC4R_Ctx (C function), 1238
 nppiResize_16u_C1R_Ctx (C function), 1238
 nppiResize_16u_C3R_Ctx (C function), 1238
 nppiResize_16u_C4R_Ctx (C function), 1238
 nppiResize_16u_P3R_Ctx (C function), 1239
 nppiResize_16u_P4R_Ctx (C function), 1239
 nppiResize_32f_AC4R_Ctx (C function), 1241
 nppiResize_32f_C1R_Ctx (C function), 1240
 nppiResize_32f_C3R_Ctx (C function), 1240
 nppiResize_32f_C4R_Ctx (C function), 1241
 nppiResize_32f_P3R_Ctx (C function), 1241
 nppiResize_32f_P4R_Ctx (C function), 1241
 nppiResizeAdvancedGetBufferHostSize_8u_C1R (C function), 1233

nppiResizeBatch_8u_AC4R_Advanced_Ctx (C function), 1246
 nppiResizeBatch_8u_AC4R_Ctx (C function), 1245
 nppiResizeBatch_8u_C1R_Advanced_Ctx (C function), 1246
 nppiResizeBatch_8u_C1R_Ctx (C function), 1245
 nppiResizeBatch_8u_C3R_Advanced_Ctx (C function), 1246
 nppiResizeBatch_8u_C3R_Ctx (C function), 1245
 nppiResizeBatch_8u_C4R_Advanced_Ctx (C function), 1246
 nppiResizeBatch_8u_C4R_Ctx (C function), 1245
 nppiResizeBatch_16f_C1R_Advanced_Ctx (C function), 1247
 nppiResizeBatch_16f_C3R_Advanced_Ctx (C function), 1247
 nppiResizeBatch_16f_C4R_Advanced_Ctx (C function), 1247
 nppiResizeBatch_32f_AC4R_Advanced_Ctx (C function), 1248
 nppiResizeBatch_32f_AC4R_Ctx (C function), 1246
 nppiResizeBatch_32f_C1R_Advanced_Ctx (C function), 1247
 nppiResizeBatch_32f_C1R_Ctx (C function), 1245
 nppiResizeBatch_32f_C3R_Advanced_Ctx (C function), 1247
 nppiResizeBatch_32f_C3R_Ctx (C function), 1245
 nppiResizeBatch_32f_C4R_Advanced_Ctx (C function), 1248
 nppiResizeBatch_32f_C4R_Ctx (C function), 1245
 NppiResizeBatchCXR (C struct), 49
 NppiResizeBatchCXR.nDstStep (C var), 49
 NppiResizeBatchCXR.nSrcStep (C var), 49
 NppiResizeBatchCXR.pDst (C var), 49
 NppiResizeBatchCXR.pSrc (C var), 49
 NppiResizeBatchROI_Advanced (C struct), 49
 NppiResizeBatchROI_Advanced.oDstRectROI (C var), 49
 NppiResizeBatchROI_Advanced.oSrcRectROI (C var), 49
 nppiResizeSqrPixel_8u_AC4R_Ctx (C function), 1228
 nppiResizeSqrPixel_8u_C1R_Advanced_Ctx (C function), 1234
 nppiResizeSqrPixel_8u_C1R_Ctx (C function), 1228
 nppiResizeSqrPixel_8u_C3R_Ctx (C function), 1228
 nppiResizeSqrPixel_8u_C4R_Ctx (C function), 1228
 nppiResizeSqrPixel_8u_P3R_Ctx (C function), 1228
 nppiResizeSqrPixel_8u_P4R_Ctx (C function), 1229
 nppiResizeSqrPixel_16s_AC4R_Ctx (C function), 1230
 nppiResizeSqrPixel_16s_C1R_Ctx (C function), 1230
 nppiResizeSqrPixel_16s_C3R_Ctx (C function), 1230
 nppiResizeSqrPixel_16s_C4R_Ctx (C function), 1230
 nppiResizeSqrPixel_16s_P3R_Ctx (C function), 1231
 nppiResizeSqrPixel_16s_P4R_Ctx (C function), 1231
 nppiResizeSqrPixel_16u_AC4R_Ctx (C function), 1229
 nppiResizeSqrPixel_16u_C1R_Ctx (C function), 1229
 nppiResizeSqrPixel_16u_C3R_Ctx (C function), 1229
 nppiResizeSqrPixel_16u_C4R_Ctx (C function), 1229
 nppiResizeSqrPixel_16u_P3R_Ctx (C function), 1230
 nppiResizeSqrPixel_16u_P4R_Ctx (C function), 1230
 nppiResizeSqrPixel_32f_AC4R_Ctx (C function), 1232
 nppiResizeSqrPixel_32f_C1R_Ctx (C function), 1231
 nppiResizeSqrPixel_32f_C3R_Ctx (C function), 1231
 nppiResizeSqrPixel_32f_C4R_Ctx (C function), 1231
 nppiResizeSqrPixel_32f_P3R_Ctx (C function), 1232
 nppiResizeSqrPixel_32f_P4R_Ctx (C function), 1232
 nppiResizeSqrPixel_64f_AC4R_Ctx (C function), 1233
 nppiResizeSqrPixel_64f_C1R_Ctx (C function), 1232
 nppiResizeSqrPixel_64f_C3R_Ctx (C function), 1232
 nppiResizeSqrPixel_64f_C4R_Ctx (C function), 1233
 nppiResizeSqrPixel_64f_P3R_Ctx (C function), 1233

nppiResizeSqrPixel_64f_P4R_Ctx (C function), 1233
 nppiRGB_8u_ToUYVP_10u_C3C3R_Ctx (C function), 528
 nppiRGB_8u_ToUYVP_10u_C3P3R_Ctx (C function), 529
 nppiRGB_8u_ToUYVP_10u_P3C3R_Ctx (C function), 529
 nppiRGB_8u_ToUYVY_8u_C3C3R_Ctx (C function), 533
 nppiRGB_8u_ToUYVY_8u_P3C3R_Ctx (C function), 533
 nppiRGB_16f_ToUYVP_10u_C3C3R_Ctx (C function), 529
 nppiRGB_16f_ToUYVP_10u_P3C3R_Ctx (C function), 529
 nppiRGB_16u_ToUYVP_10u_C3C3R_Ctx (C function), 529
 nppiRGB_16u_ToUYVP_10u_P3C3R_Ctx (C function), 529
 nppiRGB_16u_ToUYVY16_16u_C3C3R_Ctx (C function), 533
 nppiRGB_16u_ToUYVY16_16u_P3C3R_Ctx (C function), 533
 nppiRGBA_8u_ToUYVP_10u_AC4C3R_Ctx (C function), 529
 nppiRGBA_8u_ToUYVY_8u_AC4C3R_Ctx (C function), 533
 nppiRGBA_16f_ToUYVP_10u_AC4C3R_Ctx (C function), 529
 nppiRGBA_16u_ToUYVP_10u_AC4C3R_Ctx (C function), 529
 nppiRGBA_16u_ToUYVY16_16u_AC4C3R_Ctx (C function), 533
 nppiRGBToCbYCr422_8u_C3C2R_Ctx (C function), 746
 nppiRGBToCbYCr422Gamma_8u_C3C2R_Ctx (C function), 746
 nppiRGBToGray_8u_AC4C1R_Ctx (C function), 536
 nppiRGBToGray_8u_C3C1R_Ctx (C function), 536
 nppiRGBToGray_16s_AC4C1R_Ctx (C function), 538
 nppiRGBToGray_16s_C3C1R_Ctx (C function), 537
 nppiRGBToGray_16u_AC4C1R_Ctx (C function), 537
 nppiRGBToGray_16u_C3C1R_Ctx (C function), 537
 nppiRGBToGray_32f_AC4C1R_Ctx (C function), 538
 nppiRGBToGray_32f_C3C1R_Ctx (C function), 538
 nppiRGBToHLS_8u_AC4R_Ctx (C function), 781
 nppiRGBToHLS_8u_C3R_Ctx (C function), 781
 nppiRGBToHSV_8u_AC4R_Ctx (C function), 790
 nppiRGBToHSV_8u_C3R_Ctx (C function), 790
 nppiRGBToLUV_8u_AC4R_Ctx (C function), 771
 nppiRGBToLUV_8u_C3R_Ctx (C function), 770
 nppiRGBToNV12_8u_C3P2R_Ctx (C function), 535
 nppiRGBToNV12_8u_ColorTwist32f_C3P2R_Ctx (C function), 517
 nppiRGBToNV12_8u_ColorTwist32f_P3P2R_Ctx (C function), 518
 nppiRGBToNV12_8u_P3P2R_Ctx (C function), 535
 nppiRGBToNV12_16u_ColorTwist32f_C3P2R_Ctx (C function), 517
 nppiRGBToNV12_16u_ColorTwist32f_P3P2R_Ctx (C function), 518
 nppiRGBToXYZ_8u_AC4R_Ctx (C function), 768
 nppiRGBToXYZ_8u_C3R_Ctx (C function), 767
 nppiRGBToYCbCr411_JPEG_8u_C3P3R_Ctx (C function), 799
 nppiRGBToYCbCr411_JPEG_8u_P3R_Ctx (C function), 793
 nppiRGBToYCbCr420_8u_C3P3R_Ctx (C function), 751
 nppiRGBToYCbCr420_JPEG_8u_C3P3R_Ctx (C function), 799
 nppiRGBToYCbCr420_JPEG_8u_P3R_Ctx (C function), 793
 nppiRGBToYCbCr422_8u_C3C2R_Ctx (C function), 737
 nppiRGBToYCbCr422_8u_C3P3R_Ctx (C function), 737
 nppiRGBToYCbCr422_8u_P3C2R_Ctx (C function), 737
 nppiRGBToYCbCr422_JPEG_8u_C3P3R_Ctx (C function), 799
 nppiRGBToYCbCr422_JPEG_8u_P3R_Ctx (C function), 793
 nppiRGBToYCbCr444_JPEG_8u_C3P3R_Ctx (C function), 800
 nppiRGBToYCbCr444_JPEG_8u_P3R_Ctx (C function), 794
 nppiRGBToYCbCr_8u_AC4P3R_Ctx (C function), 727
 nppiRGBToYCbCr_8u_AC4R_Ctx (C function), 726
 nppiRGBToYCbCr_8u_C3P3R_Ctx (C function), 727
 nppiRGBToYCbCr_8u_C3R_Ctx (C function), 726
 nppiRGBToYCbCr_8u_P3R_Ctx (C function), 726
 nppiRGBToYCC_8u_AC4R_Ctx (C function), 776
 nppiRGBToYCC_8u_C3R_Ctx (C function), 775

nppiRGBToYCrCb420_8u_AC4P3R_Ctx (C function), 753
 nppiRGBToYCrCb422_8u_C3C2R_Ctx (C function), 741
 nppiRGBToYCrCb422_8u_P3C2R_Ctx (C function), 741
 nppiRGBToYUV420_8u_C3P3R_Ctx (C function), 714
 nppiRGBToYUV420_8u_ColorTwist32f_C3P3R_Ctx (C function), 513
 nppiRGBToYUV420_8u_ColorTwist32f_P3R_Ctx (C function), 512
 nppiRGBToYUV420_8u_P3R_Ctx (C function), 714
 nppiRGBToYUV420_16u_ColorTwist32f_C3P3R_Ctx (C function), 514
 nppiRGBToYUV420_16u_ColorTwist32f_P3R_Ctx (C function), 513
 nppiRGBToYUV422_8u_C3C2R_Ctx (C function), 708
 nppiRGBToYUV422_8u_C3P3R_Ctx (C function), 708
 nppiRGBToYUV422_8u_ColorTwist32f_C3C2R_Ctx (C function), 515
 nppiRGBToYUV422_8u_ColorTwist32f_C3P3R_Ctx (C function), 516
 nppiRGBToYUV422_8u_ColorTwist32f_P3R_Ctx (C function), 514
 nppiRGBToYUV422_8u_P3R_Ctx (C function), 708
 nppiRGBToYUV422_16u_ColorTwist32f_C3C2R_Ctx (C function), 516
 nppiRGBToYUV422_16u_ColorTwist32f_C3P3R_Ctx (C function), 516
 nppiRGBToYUV422_16u_ColorTwist32f_P3R_Ctx (C function), 515
 nppiRGBToYUV_8u_AC4P4R_Ctx (C function), 696
 nppiRGBToYUV_8u_AC4R_Ctx (C function), 695
 nppiRGBToYUV_8u_C3P3R_Ctx (C function), 695
 nppiRGBToYUV_8u_C3R_Ctx (C function), 694
 nppiRGBToYUV_8u_P3R_Ctx (C function), 695
 nppiRotate_8u_AC4R_Ctx (C function), 1257
 nppiRotate_8u_C1R_Ctx (C function), 1257
 nppiRotate_8u_C3R_Ctx (C function), 1257
 nppiRotate_8u_C4R_Ctx (C function), 1257
 nppiRotate_16u_AC4R_Ctx (C function), 1258
 nppiRotate_16u_C1R_Ctx (C function), 1258
 nppiRotate_16u_C3R_Ctx (C function), 1258
 nppiRotate_16u_C4R_Ctx (C function), 1258
 nppiRotate_32f_AC4R_Ctx (C function), 1259
 nppiRotate_32f_C1R_Ctx (C function), 1258
 nppiRotate_32f_C3R_Ctx (C function), 1258
 nppiRotate_32f_C4R_Ctx (C function), 1258
 nppiRShiftC_8s_AC4IR_Ctx (C function), 443
 nppiRShiftC_8s_AC4R_Ctx (C function), 442
 nppiRShiftC_8s_C1IR_Ctx (C function), 441
 nppiRShiftC_8s_C1R_Ctx (C function), 441
 nppiRShiftC_8s_C3IR_Ctx (C function), 442
 nppiRShiftC_8s_C3R_Ctx (C function), 442
 nppiRShiftC_8s_C4IR_Ctx (C function), 443
 nppiRShiftC_8s_C4R_Ctx (C function), 443
 nppiRShiftC_8u_AC4IR_Ctx (C function), 440
 nppiRShiftC_8u_AC4R_Ctx (C function), 440
 nppiRShiftC_8u_C1IR_Ctx (C function), 439
 nppiRShiftC_8u_C1R_Ctx (C function), 438
 nppiRShiftC_8u_C3IR_Ctx (C function), 439
 nppiRShiftC_8u_C3R_Ctx (C function), 439
 nppiRShiftC_8u_C4IR_Ctx (C function), 441
 nppiRShiftC_8u_C4R_Ctx (C function), 440
 nppiRShiftC_16s_AC4IR_Ctx (C function), 448
 nppiRShiftC_16s_AC4R_Ctx (C function), 448
 nppiRShiftC_16s_C1IR_Ctx (C function), 447
 nppiRShiftC_16s_C1R_Ctx (C function), 447
 nppiRShiftC_16s_C3IR_Ctx (C function), 448
 nppiRShiftC_16s_C3R_Ctx (C function), 447
 nppiRShiftC_16s_C4IR_Ctx (C function), 449
 nppiRShiftC_16s_C4R_Ctx (C function), 449
 nppiRShiftC_16u_AC4IR_Ctx (C function), 446
 nppiRShiftC_16u_AC4R_Ctx (C function), 445
 nppiRShiftC_16u_C1IR_Ctx (C function), 444
 nppiRShiftC_16u_C1R_Ctx (C function), 444
 nppiRShiftC_16u_C3IR_Ctx (C function), 445
 nppiRShiftC_16u_C3R_Ctx (C function), 445
 nppiRShiftC_16u_C4IR_Ctx (C function), 446
 nppiRShiftC_16u_C4R_Ctx (C function), 446
 nppiRShiftC_32s_AC4IR_Ctx (C function), 451
 nppiRShiftC_32s_AC4R_Ctx (C function), 451
 nppiRShiftC_32s_C1IR_Ctx (C function), 450
 nppiRShiftC_32s_C1R_Ctx (C function), 449
 nppiRShiftC_32s_C3IR_Ctx (C function), 450
 nppiRShiftC_32s_C3R_Ctx (C function), 450
 nppiRShiftC_32s_C4IR_Ctx (C function), 452
 nppiRShiftC_32s_C4R_Ctx (C function), 452
 nppiSameNormLevelGetBufferHostSize_8s32f_AC4R_Ctx (C function), 1547
 nppiSameNormLevelGetBufferHostSize_8s32f_C1R_Ctx (C function), 1547
 nppiSameNormLevelGetBufferHostSize_8s32f_C3R_Ctx (C function), 1547
 nppiSameNormLevelGetBufferHostSize_8s32f_C4R_Ctx (C function), 1547
 nppiSameNormLevelGetBufferHostSize_8u32f_AC4R_Ctx (C function), 1547
 nppiSameNormLevelGetBufferHostSize_8u32f_C1R_Ctx (C function), 1546
 nppiSameNormLevelGetBufferHostSize_8u32f_C3R_Ctx (C function), 1546

nppiSameNormLevelGetBufferHostSize_8u32f_Ac4R_Ctx (C function), 873
 (C function), 1546
 nppiSameNormLevelGetBufferHostSize_8u_AC4R_Ctx_16u8u_C3R_Ctx (C function), 873
 (C function), 1545
 nppiSameNormLevelGetBufferHostSize_8u_C1R_Ac4R_Ctx_32f8u_AC4R_Ctx (C function), 875
 (C function), 1544
 nppiSameNormLevelGetBufferHostSize_8u_C3R_Ac4R_Ctx_32f8u_C3R_Ctx (C function), 875
 (C function), 1544
 nppiSameNormLevelGetBufferHostSize_8u_C4R_Ac4R_Ctx_32s8u_AC4R_Ctx (C function), 874
 (C function), 1545
 nppiSameNormLevelGetBufferHostSize_16u32f_Ac4R_Ctx_32s8u_C3R_Ctx (C function), 874
 (C function), 1548
 nppiSameNormLevelGetBufferHostSize_16u32f_Ac4R_Ctx_32s8u_C4R_Ctx (C function), 874
 (C function), 1547
 nppiSameNormLevelGetBufferHostSize_16u32f_Ac4R_Ctx_32s8u_C1R_Ctx (C function), 1223
 (C function), 1547
 nppiSameNormLevelGetBufferHostSize_16u32f_Ac4R_Ctx_32s8u_C3R_Ctx (C function), 1221
 (C function), 1548
 nppiSameNormLevelGetBufferHostSize_32f_Ac4R_Ctx_32s8u_C1R_Ctx (C function), 1222
 (C function), 1545
 nppiSameNormLevelGetBufferHostSize_32f_C1R_Ac4R_Ctx_8s_AC4R_Ctx (C function), 806
 (C function), 1545
 nppiSameNormLevelGetBufferHostSize_32f_C1R_Ac4R_Ctx_8s_C2R_Ctx (C function), 806
 (C function), 1545
 nppiSameNormLevelGetBufferHostSize_32f_C1R_Ac4R_Ctx_8s_C3R_Ctx (C function), 806
 (C function), 1545
 nppiSameNormLevelGetBufferHostSize_32f_C1R_Ac4R_Ctx_8s_C4R_Ctx (C function), 806
 (C function), 1545
 nppiSameNormLevelGetBufferHostSize_64f_Ac4R_Ctx_8u_AC4R_Ctx (C function), 806
 (C function), 1546
 nppiSameNormLevelGetBufferHostSize_64f_C1R_Ac4R_Ctx_8u_C1MR_Ctx (C function), 812
 (C function), 1545
 nppiSameNormLevelGetBufferHostSize_64f_C1R_Ac4R_Ctx_8u_C2R_Ctx (C function), 806
 (C function), 1545
 nppiSameNormLevelGetBufferHostSize_64f_C1R_Ac4R_Ctx_8u_C3CR_Ctx (C function), 815
 (C function), 1546
 nppiSameNormLevelGetBufferHostSize_64f_C1R_Ac4R_Ctx_8u_C3MR_Ctx (C function), 812
 (C function), 1546
 nppiSameNormLevelGetBufferHostSize_64f_C1R_Ac4R_Ctx_8u_C4CR_Ctx (C function), 815
 (C function), 1546
 nppiScale_8u16s_AC4R_Ctx (C function), 871
 nppiScale_8u16s_C1R_Ctx (C function), 870
 nppiScale_8u16s_C3R_Ctx (C function), 870
 nppiScale_8u16s_C4R_Ctx (C function), 870
 nppiScale_8u16u_AC4R_Ctx (C function), 870
 nppiScale_8u16u_C1R_Ctx (C function), 870
 nppiScale_8u16u_C3R_Ctx (C function), 870
 nppiScale_8u16u_C4R_Ctx (C function), 870
 nppiScale_8u32f_AC4R_Ctx (C function), 872
 nppiScale_8u32f_C1R_Ctx (C function), 871
 nppiScale_8u32f_C3R_Ctx (C function), 871
 nppiScale_8u32f_C4R_Ctx (C function), 872
 nppiScale_8u32s_AC4R_Ctx (C function), 871
 nppiScale_8u32s_C1R_Ctx (C function), 871
 nppiScale_8u32s_C3R_Ctx (C function), 871
 nppiScale_8u32s_C4R_Ctx (C function), 871
 nppiScale_16s8u_AC4R_Ctx (C function), 874
 nppiScale_16s8u_C1R_Ctx (C function), 873
 nppiScale_16s8u_C3R_Ctx (C function), 874
 nppiScale_16s8u_C4R_Ctx (C function), 874
 nppiScale_16u8u_AC4R_Ctx (C function), 873
 nppiScale_16u8u_C1R_Ctx (C function), 873
 nppiScale_16u8u_C3R_Ctx (C function), 873
 nppiScale_16u8u_C4R_Ctx (C function), 873
 nppiScale_32f8u_C1R_Ctx (C function), 874
 nppiScale_32f8u_C3R_Ctx (C function), 875
 nppiScale_32f8u_C4R_Ctx (C function), 875
 nppiScale_32s8u_C1R_Ctx (C function), 874
 nppiScale_32s8u_C4R_Ctx (C function), 874
 nppiSegmentWatershed_8u_C1IR_Ctx (C function), 1222
 (C function), 1223
 nppiSegmentWatershed_16u_C1IR_Ctx (C function), 1223
 (C function), 1221
 nppiSegmentWatershedGetBufferSize_8u_C1R_Ctx (C function), 1222
 (C function), 1222
 nppiSet_8s_AC4R_Ctx (C function), 806
 nppiSet_8s_C1R_Ctx (C function), 806
 nppiSet_8s_C2R_Ctx (C function), 806
 nppiSet_8s_C3R_Ctx (C function), 806
 nppiSet_8s_C4R_Ctx (C function), 806
 nppiSet_8u_AC4MR_Ctx (C function), 812
 nppiSet_8u_C1MR_Ctx (C function), 812
 nppiSet_8u_C2R_Ctx (C function), 806
 nppiSet_8u_C3CR_Ctx (C function), 815
 nppiSet_8u_C3MR_Ctx (C function), 812
 nppiSet_8u_C3R_Ctx (C function), 806
 nppiSet_8u_C4CR_Ctx (C function), 815
 nppiSet_8u_C4MR_Ctx (C function), 812
 nppiSet_8u_C4R_Ctx (C function), 806
 nppiSet_16f_C1R_Ctx (C function), 809
 nppiSet_16f_C2R_Ctx (C function), 810
 nppiSet_16f_C3R_Ctx (C function), 810
 nppiSet_16f_C4R_Ctx (C function), 810
 nppiSet_16s_AC4MR_Ctx (C function), 813
 nppiSet_16s_AC4R_Ctx (C function), 807
 nppiSet_16s_C1MR_Ctx (C function), 813
 nppiSet_16s_C1R_Ctx (C function), 807
 nppiSet_16s_C2R_Ctx (C function), 807
 nppiSet_16s_C3CR_Ctx (C function), 815
 nppiSet_16s_C3MR_Ctx (C function), 813
 nppiSet_16s_C3R_Ctx (C function), 807
 nppiSet_16s_C4CR_Ctx (C function), 815
 nppiSet_16s_C4MR_Ctx (C function), 813
 nppiSet_16s_C4R_Ctx (C function), 807
 nppiSet_16sc_AC4R_Ctx (C function), 808
 nppiSet_16sc_C1R_Ctx (C function), 807
 nppiSet_16sc_C2R_Ctx (C function), 808

nppiSet_16sc_C3R_Ctx (C function), 808
 nppiSet_16sc_C4R_Ctx (C function), 808
 nppiSet_16u_AC4MR_Ctx (C function), 813
 nppiSet_16u_AC4R_Ctx (C function), 807
 nppiSet_16u_C1MR_Ctx (C function), 812
 nppiSet_16u_C1R_Ctx (C function), 807
 nppiSet_16u_C2R_Ctx (C function), 807
 nppiSet_16u_C3CR_Ctx (C function), 815
 nppiSet_16u_C3MR_Ctx (C function), 812
 nppiSet_16u_C3R_Ctx (C function), 807
 nppiSet_16u_C4CR_Ctx (C function), 815
 nppiSet_16u_C4MR_Ctx (C function), 812
 nppiSet_16u_C4R_Ctx (C function), 807
 nppiSet_32f_AC4MR_Ctx (C function), 814
 nppiSet_32f_AC4R_Ctx (C function), 810
 nppiSet_32f_C1MR_Ctx (C function), 814
 nppiSet_32f_C1R_Ctx (C function), 810
 nppiSet_32f_C2R (C function), 810
 nppiSet_32f_C2R_Ctx (C function), 810
 nppiSet_32f_C3CR_Ctx (C function), 816
 nppiSet_32f_C3MR_Ctx (C function), 814
 nppiSet_32f_C3R_Ctx (C function), 810
 nppiSet_32f_C4CR_Ctx (C function), 816
 nppiSet_32f_C4MR_Ctx (C function), 814
 nppiSet_32f_C4R_Ctx (C function), 810
 nppiSet_32fc_AC4R_Ctx (C function), 811
 nppiSet_32fc_C1R_Ctx (C function), 810
 nppiSet_32fc_C2R_Ctx (C function), 811
 nppiSet_32fc_C3R_Ctx (C function), 811
 nppiSet_32fc_C4R_Ctx (C function), 811
 nppiSet_32s_AC4MR_Ctx (C function), 814
 nppiSet_32s_AC4R_Ctx (C function), 808
 nppiSet_32s_C1MR_Ctx (C function), 813
 nppiSet_32s_C1R_Ctx (C function), 808
 nppiSet_32s_C2R_Ctx (C function), 808
 nppiSet_32s_C3CR_Ctx (C function), 815
 nppiSet_32s_C3MR_Ctx (C function), 813
 nppiSet_32s_C3R_Ctx (C function), 808
 nppiSet_32s_C4CR_Ctx (C function), 816
 nppiSet_32s_C4MR_Ctx (C function), 813
 nppiSet_32s_C4R_Ctx (C function), 808
 nppiSet_32sc_AC4R_Ctx (C function), 809
 nppiSet_32sc_C1R_Ctx (C function), 809
 nppiSet_32sc_C2R_Ctx (C function), 809
 nppiSet_32sc_C3R_Ctx (C function), 809
 nppiSet_32sc_C4R_Ctx (C function), 809
 nppiSet_32u_AC4R_Ctx (C function), 809
 nppiSet_32u_C1R_Ctx (C function), 808
 nppiSet_32u_C2R_Ctx (C function), 809
 nppiSet_32u_C3R_Ctx (C function), 809
 nppiSet_32u_C4R_Ctx (C function), 809
 nppiSignedDistanceTransformAbsPBA_32f64f_C1R_Ctx (C function), 1162
 nppiSignedDistanceTransformAbsPBA_32f_C1R_Ctx (C function), 1159
 nppiSignedDistanceTransformAbsPBA_64f_C1R_Ctx (C function), 1165
 nppiSignedDistanceTransformPBA_32f64f_C1R_Ctx (C function), 1160
 nppiSignedDistanceTransformPBA_32f_C1R_Ctx (C function), 1157
 nppiSignedDistanceTransformPBA_64f_C1R_Ctx (C function), 1163
 nppiSignedDistanceTransformPBAGet64fBufferSize (C function), 1125
 nppiSignedDistanceTransformPBAGetAntialiasingBufferSize (C function), 1126
 nppiSignedDistanceTransformPBAGetBufferSize (C function), 1125
 NppiSize (C struct), 34
 NppiSize.height (C var), 35
 NppiSize.width (C var), 35
 nppiSqr_8u_AC4IRSfs_Ctx (C function), 378
 nppiSqr_8u_AC4RSfs_Ctx (C function), 377
 nppiSqr_8u_C1IRSfs_Ctx (C function), 376
 nppiSqr_8u_C1RSfs_Ctx (C function), 376
 nppiSqr_8u_C3IRSfs_Ctx (C function), 377
 nppiSqr_8u_C3RSfs_Ctx (C function), 376
 nppiSqr_8u_C4IRSfs_Ctx (C function), 378
 nppiSqr_8u_C4RSfs_Ctx (C function), 378
 nppiSqr_16f_C1IR_Ctx (C function), 385
 nppiSqr_16f_C1R_Ctx (C function), 384
 nppiSqr_16f_C3IR_Ctx (C function), 385
 nppiSqr_16f_C3R_Ctx (C function), 385
 nppiSqr_16f_C4IR_Ctx (C function), 386
 nppiSqr_16f_C4R_Ctx (C function), 386
 nppiSqr_16s_AC4IRSfs_Ctx (C function), 383
 nppiSqr_16s_AC4RSfs_Ctx (C function), 383
 nppiSqr_16s_C1IRSfs_Ctx (C function), 382
 nppiSqr_16s_C1RSfs_Ctx (C function), 382
 nppiSqr_16s_C3IRSfs_Ctx (C function), 383
 nppiSqr_16s_C3RSfs_Ctx (C function), 382
 nppiSqr_16s_C4IRSfs_Ctx (C function), 384
 nppiSqr_16s_C4RSfs_Ctx (C function), 384
 nppiSqr_16u_AC4IRSfs_Ctx (C function), 380
 nppiSqr_16u_AC4RSfs_Ctx (C function), 380
 nppiSqr_16u_C1IRSfs_Ctx (C function), 379
 nppiSqr_16u_C1RSfs_Ctx (C function), 379
 nppiSqr_16u_C3IRSfs_Ctx (C function), 380
 nppiSqr_16u_C3RSfs_Ctx (C function), 379
 nppiSqr_16u_C4IRSfs_Ctx (C function), 381
 nppiSqr_16u_C4RSfs_Ctx (C function), 381
 nppiSqr_32f_AC4IR_Ctx (C function), 388
 nppiSqr_32f_AC4R_Ctx (C function), 387
 nppiSqr_32f_C1IR_Ctx (C function), 387
 nppiSqr_32f_C1R_Ctx (C function), 386
 nppiSqr_32f_C3IR_Ctx (C function), 387

nppiSqr_32f_C3R_Ctx (C function), 387
 nppiSqr_32f_C4IR_Ctx (C function), 388
 nppiSqr_32f_C4R_Ctx (C function), 388
 nppiSqrDistanceFull_Norm_8s32f_AC4R_Ctx (C function), 1505
 nppiSqrDistanceFull_Norm_8s32f_C1R_Ctx (C function), 1505
 nppiSqrDistanceFull_Norm_8s32f_C3R_Ctx (C function), 1505
 nppiSqrDistanceFull_Norm_8s32f_C4R_Ctx (C function), 1505
 nppiSqrDistanceFull_Norm_8u32f_AC4R_Ctx (C function), 1505
 nppiSqrDistanceFull_Norm_8u32f_C1R_Ctx (C function), 1504
 nppiSqrDistanceFull_Norm_8u32f_C3R_Ctx (C function), 1504
 nppiSqrDistanceFull_Norm_8u32f_C4R_Ctx (C function), 1504
 nppiSqrDistanceFull_Norm_8u_AC4RSfs_Ctx (C function), 1503
 nppiSqrDistanceFull_Norm_8u_C1RSfs_Ctx (C function), 1503
 nppiSqrDistanceFull_Norm_8u_C3RSfs_Ctx (C function), 1503
 nppiSqrDistanceFull_Norm_8u_C4RSfs_Ctx (C function), 1503
 nppiSqrDistanceFull_Norm_16u32f_AC4R_Ctx (C function), 1506
 nppiSqrDistanceFull_Norm_16u32f_C1R_Ctx (C function), 1506
 nppiSqrDistanceFull_Norm_16u32f_C3R_Ctx (C function), 1506
 nppiSqrDistanceFull_Norm_16u32f_C4R_Ctx (C function), 1506
 nppiSqrDistanceFull_Norm_32f_AC4R_Ctx (C function), 1504
 nppiSqrDistanceFull_Norm_32f_C1R_Ctx (C function), 1503
 nppiSqrDistanceFull_Norm_32f_C3R_Ctx (C function), 1504
 nppiSqrDistanceFull_Norm_32f_C4R_Ctx (C function), 1504
 nppiSqrDistanceSame_Norm_8s32f_AC4R_Ctx (C function), 1510
 nppiSqrDistanceSame_Norm_8s32f_C1R_Ctx (C function), 1509
 nppiSqrDistanceSame_Norm_8s32f_C3R_Ctx (C function), 1509
 nppiSqrDistanceSame_Norm_8s32f_C4R_Ctx (C function), 1509
 nppiSqrDistanceSame_Norm_8u32f_AC4R_Ctx (C function), 1509
 nppiSqrDistanceSame_Norm_8u32f_C1R_Ctx (C function), 1508
 nppiSqrDistanceSame_Norm_8u32f_C3R_Ctx (C function), 1508
 nppiSqrDistanceSame_Norm_8u32f_C4R_Ctx (C function), 1508
 nppiSqrDistanceSame_Norm_8u_AC4RSfs_Ctx (C function), 1507
 nppiSqrDistanceSame_Norm_8u_C1RSfs_Ctx (C function), 1507
 nppiSqrDistanceSame_Norm_8u_C3RSfs_Ctx (C function), 1507
 nppiSqrDistanceSame_Norm_8u_C4RSfs_Ctx (C function), 1507
 nppiSqrDistanceSame_Norm_16u32f_AC4R (C function), 1510
 nppiSqrDistanceSame_Norm_16u32f_AC4R_Ctx (C function), 1510
 nppiSqrDistanceSame_Norm_16u32f_C1R_Ctx (C function), 1510
 nppiSqrDistanceSame_Norm_16u32f_C3R_Ctx (C function), 1510
 nppiSqrDistanceSame_Norm_16u32f_C4R_Ctx (C function), 1510
 nppiSqrDistanceSame_Norm_32f_AC4R_Ctx (C function), 1508
 nppiSqrDistanceSame_Norm_32f_C1R_Ctx (C function), 1507
 nppiSqrDistanceSame_Norm_32f_C3R_Ctx (C function), 1508
 nppiSqrDistanceSame_Norm_32f_C4R_Ctx (C function), 1508
 nppiSqrDistanceValid_Norm_8s32f_AC4R_Ctx (C function), 1514
 nppiSqrDistanceValid_Norm_8s32f_C1R_Ctx (C function), 1513
 nppiSqrDistanceValid_Norm_8s32f_C3R_Ctx (C function), 1513
 nppiSqrDistanceValid_Norm_8s32f_C4R_Ctx (C function), 1514
 nppiSqrDistanceValid_Norm_8u32f_AC4R_Ctx (C function), 1513
 nppiSqrDistanceValid_Norm_8u32f_C1R_Ctx (C function), 1512
 nppiSqrDistanceValid_Norm_8u32f_C3R_Ctx (C function), 1513
 nppiSqrDistanceValid_Norm_8u32f_C4R_Ctx (C function), 1513
 nppiSqrDistanceValid_Norm_8u_AC4RSfs_Ctx (C function), 1511
 nppiSqrDistanceValid_Norm_8u_C1RSfs_Ctx (C function), 1511
 nppiSqrDistanceValid_Norm_8u_C3RSfs_Ctx (C function), 1511
 nppiSqrDistanceValid_Norm_8u_C4RSfs_Ctx (C function), 1511

(C function), 1511
 nppiSqrDistanceValid_Norm_16u32f_AC4R_Ctx (C function), 1515
 nppiSqrDistanceValid_Norm_16u32f_C1R_Ctx (C function), 1514
 nppiSqrDistanceValid_Norm_16u32f_C3R_Ctx (C function), 1514
 nppiSqrDistanceValid_Norm_16u32f_C4R_Ctx (C function), 1514
 nppiSqrDistanceValid_Norm_32f_AC4R_Ctx (C function), 1512
 nppiSqrDistanceValid_Norm_32f_C1R_Ctx (C function), 1512
 nppiSqrDistanceValid_Norm_32f_C3R_Ctx (C function), 1512
 nppiSqrDistanceValid_Norm_32f_C4R_Ctx (C function), 1512
 nppiSqrIntegral_8u32f64f_C1R_Ctx (C function), 1486
 nppiSqrIntegral_8u32s64f_C1R_Ctx (C function), 1486
 nppiSqrIntegral_8u32s_C1R_Ctx (C function), 1485
 nppiSqrt_8u_AC4IRSfs_Ctx (C function), 391
 nppiSqrt_8u_AC4RSfs_Ctx (C function), 390
 nppiSqrt_8u_C1IRSfs_Ctx (C function), 389
 nppiSqrt_8u_C1RSfs_Ctx (C function), 389
 nppiSqrt_8u_C3IRSfs_Ctx (C function), 390
 nppiSqrt_8u_C3RSfs_Ctx (C function), 390
 nppiSqrt_16f_C1IR_Ctx (C function), 396
 nppiSqrt_16f_C1R_Ctx (C function), 395
 nppiSqrt_16f_C3IR_Ctx (C function), 396
 nppiSqrt_16f_C3R_Ctx (C function), 396
 nppiSqrt_16f_C4IR_Ctx (C function), 397
 nppiSqrt_16f_C4R_Ctx (C function), 397
 nppiSqrt_16s_AC4IRSfs_Ctx (C function), 395
 nppiSqrt_16s_AC4RSfs_Ctx (C function), 395
 nppiSqrt_16s_C1IRSfs_Ctx (C function), 394
 nppiSqrt_16s_C1RSfs_Ctx (C function), 393
 nppiSqrt_16s_C3IRSfs_Ctx (C function), 394
 nppiSqrt_16s_C3RSfs_Ctx (C function), 394
 nppiSqrt_16u_AC4IRSfs_Ctx (C function), 393
 nppiSqrt_16u_AC4RSfs_Ctx (C function), 392
 nppiSqrt_16u_C1IRSfs_Ctx (C function), 391
 nppiSqrt_16u_C1RSfs_Ctx (C function), 391
 nppiSqrt_16u_C3IRSfs_Ctx (C function), 392
 nppiSqrt_16u_C3RSfs_Ctx (C function), 392
 nppiSqrt_32f_AC4IR_Ctx (C function), 399
 nppiSqrt_32f_AC4R_Ctx (C function), 398
 nppiSqrt_32f_C1IR_Ctx (C function), 397
 nppiSqrt_32f_C1R_Ctx (C function), 397
 nppiSqrt_32f_C3IR_Ctx (C function), 398
 nppiSqrt_32f_C3R_Ctx (C function), 398
 nppiSqrt_32f_C4IR_Ctx (C function), 399
 nppiSqrt_32f_C4R_Ctx (C function), 399
 nppiSSIM_8u_C1R_Ctx (C function), 1642
 nppiSSIM_8u_C3R_Ctx (C function), 1642
 nppiSSIMBatch_8u_C1R_Ctx (C function), 1649
 nppiSSIMBatch_8u_C3R_Ctx (C function), 1649
 nppiSSIMBatchGetBufferHostSize_8u_C1R_Ctx (C function), 1653
 nppiSSIMBatchGetBufferHostSize_8u_C3R_Ctx (C function), 1653
 nppiSSIMGetBufferHostSize_8u_C1R_Ctx (C function), 1645
 nppiSSIMGetBufferHostSize_8u_C3R_Ctx (C function), 1645
 nppiSub_8u_AC4IRSfs_Ctx (C function), 302
 nppiSub_8u_AC4RSfs_Ctx (C function), 302
 nppiSub_8u_C1IRSfs_Ctx (C function), 300
 nppiSub_8u_C1RSfs_Ctx (C function), 300
 nppiSub_8u_C3IRSfs_Ctx (C function), 301
 nppiSub_8u_C3RSfs_Ctx (C function), 301
 nppiSub_8u_C4IRSfs_Ctx (C function), 303
 nppiSub_8u_C4RSfs_Ctx (C function), 303
 nppiSub_16f_C1IR_Ctx (C function), 319
 nppiSub_16f_C1R_Ctx (C function), 319
 nppiSub_16f_C3IR_Ctx (C function), 320
 nppiSub_16f_C3R_Ctx (C function), 319
 nppiSub_16f_C4IR_Ctx (C function), 321
 nppiSub_16f_C4R_Ctx (C function), 320
 nppiSub_16s_AC4IRSfs_Ctx (C function), 309
 nppiSub_16s_AC4RSfs_Ctx (C function), 309
 nppiSub_16s_C1IRSfs_Ctx (C function), 307
 nppiSub_16s_C1RSfs_Ctx (C function), 307
 nppiSub_16s_C3IRSfs_Ctx (C function), 308
 nppiSub_16s_C3RSfs_Ctx (C function), 308
 nppiSub_16s_C4IRSfs_Ctx (C function), 310
 nppiSub_16s_C4RSfs_Ctx (C function), 310
 nppiSub_16sc_AC4IRSfs_Ctx (C function), 313
 nppiSub_16sc_AC4RSfs_Ctx (C function), 312
 nppiSub_16sc_C1IRSfs_Ctx (C function), 311
 nppiSub_16sc_C1RSfs_Ctx (C function), 310
 nppiSub_16sc_C3IRSfs_Ctx (C function), 312
 nppiSub_16sc_C3RSfs_Ctx (C function), 311
 nppiSub_16u_AC4IRSfs_Ctx (C function), 306
 nppiSub_16u_AC4RSfs_Ctx (C function), 305
 nppiSub_16u_C1IRSfs_Ctx (C function), 304
 nppiSub_16u_C1RSfs_Ctx (C function), 303
 nppiSub_16u_C3IRSfs_Ctx (C function), 305
 nppiSub_16u_C3RSfs_Ctx (C function), 304
 nppiSub_16u_C4IRSfs_Ctx (C function), 306
 nppiSub_16u_C4RSfs_Ctx (C function), 306
 nppiSub_32f_AC4IR_Ctx (C function), 323
 nppiSub_32f_AC4R_Ctx (C function), 322
 nppiSub_32f_C1IR_Ctx (C function), 321
 nppiSub_32f_C1R_Ctx (C function), 321
 nppiSub_32f_C3IR_Ctx (C function), 322

nppiSub_32f_C3R_Ctx (C function), 322
 nppiSub_32f_C4IR_Ctx (C function), 324
 nppiSub_32f_C4R_Ctx (C function), 323
 nppiSub_32fc_AC4IR_Ctx (C function), 326
 nppiSub_32fc_AC4R_Ctx (C function), 326
 nppiSub_32fc_C1IR_Ctx (C function), 324
 nppiSub_32fc_C1R_Ctx (C function), 324
 nppiSub_32fc_C3IR_Ctx (C function), 325
 nppiSub_32fc_C3R_Ctx (C function), 325
 nppiSub_32fc_C4IR_Ctx (C function), 327
 nppiSub_32fc_C4R_Ctx (C function), 326
 nppiSub_32s_C1IRSfs_Ctx (C function), 314
 nppiSub_32s_C1R_Ctx (C function), 313
 nppiSub_32s_C1RSfs_Ctx (C function), 313
 nppiSub_32s_C3IRSfs_Ctx (C function), 315
 nppiSub_32s_C3RSfs_Ctx (C function), 314
 nppiSub_32s_C4IRSfs_Ctx (C function), 316
 nppiSub_32s_C4RSfs_Ctx (C function), 315
 nppiSub_32sc_AC4IRSfs_Ctx (C function), 318
 nppiSub_32sc_AC4RSfs_Ctx (C function), 318
 nppiSub_32sc_C1IRSfs_Ctx (C function), 317
 nppiSub_32sc_C1RSfs_Ctx (C function), 316
 nppiSub_32sc_C3IRSfs_Ctx (C function), 317
 nppiSub_32sc_C3RSfs_Ctx (C function), 317
 nppiSubC_8u_AC4IRSfs_Ctx (C function), 150
 nppiSubC_8u_AC4RSfs_Ctx (C function), 149
 nppiSubC_8u_C1IRSfs_Ctx (C function), 147
 nppiSubC_8u_C1RSfs_Ctx (C function), 146
 nppiSubC_8u_C3IRSfs_Ctx (C function), 149
 nppiSubC_8u_C3RSfs_Ctx (C function), 148
 nppiSubC_8u_C4IRSfs_Ctx (C function), 152
 nppiSubC_8u_C4RSfs_Ctx (C function), 151
 nppiSubC_16f_C1IR_Ctx (C function), 175
 nppiSubC_16f_C1R_Ctx (C function), 174
 nppiSubC_16f_C3IR_Ctx (C function), 176
 nppiSubC_16f_C3R_Ctx (C function), 175
 nppiSubC_16f_C4IR_Ctx (C function), 177
 nppiSubC_16f_C4R_Ctx (C function), 177
 nppiSubC_16s_AC4IRSfs_Ctx (C function), 163
 nppiSubC_16s_AC4RSfs_Ctx (C function), 162
 nppiSubC_16s_C1IRSfs_Ctx (C function), 160
 nppiSubC_16s_C1RSfs_Ctx (C function), 159
 nppiSubC_16s_C3IRSfs_Ctx (C function), 162
 nppiSubC_16s_C3RSfs_Ctx (C function), 161
 nppiSubC_16s_C4IRSfs_Ctx (C function), 165
 nppiSubC_16s_C4RSfs_Ctx (C function), 164
 nppiSubC_16sc_AC4IRSfs_Ctx (C function), 168
 nppiSubC_16sc_AC4RSfs_Ctx (C function), 167
 nppiSubC_16sc_C1IRSfs_Ctx (C function), 166
 nppiSubC_16sc_C1RSfs_Ctx (C function), 166
 nppiSubC_16sc_C3IRSfs_Ctx (C function), 167
 nppiSubC_16sc_C3RSfs_Ctx (C function), 167
 nppiSubC_16u_AC4IRSfs_Ctx (C function), 157
 nppiSubC_16u_AC4RSfs_Ctx (C function), 156
 nppiSubC_16u_C1IRSfs_Ctx (C function), 153
 nppiSubC_16u_C1RSfs_Ctx (C function), 153
 nppiSubC_16u_C3IRSfs_Ctx (C function), 155
 nppiSubC_16u_C3RSfs_Ctx (C function), 154
 nppiSubC_16u_C4IRSfs_Ctx (C function), 158
 nppiSubC_16u_C4RSfs_Ctx (C function), 158
 nppiSubC_32f_AC4IR_Ctx (C function), 182
 nppiSubC_32f_AC4R_Ctx (C function), 181
 nppiSubC_32f_C1IR_Ctx (C function), 179
 nppiSubC_32f_C1R_Ctx (C function), 178
 nppiSubC_32f_C3IR_Ctx (C function), 180
 nppiSubC_32f_C3R_Ctx (C function), 180
 nppiSubC_32f_C4IR_Ctx (C function), 183
 nppiSubC_32f_C4R_Ctx (C function), 182
 nppiSubC_32fc_AC4IR_Ctx (C function), 186
 nppiSubC_32fc_AC4R_Ctx (C function), 185
 nppiSubC_32fc_C1IR_Ctx (C function), 184
 nppiSubC_32fc_C1R_Ctx (C function), 184
 nppiSubC_32fc_C3IR_Ctx (C function), 185
 nppiSubC_32fc_C3R_Ctx (C function), 185
 nppiSubC_32fc_C4IR_Ctx (C function), 186
 nppiSubC_32fc_C4R_Ctx (C function), 186
 nppiSubC_32s_C1IRSfs_Ctx (C function), 169
 nppiSubC_32s_C1RSfs_Ctx (C function), 168
 nppiSubC_32s_C3IRSfs_Ctx (C function), 171
 nppiSubC_32s_C3RSfs_Ctx (C function), 170
 nppiSubC_32sc_AC4IRSfs_Ctx (C function), 173
 nppiSubC_32sc_AC4RSfs_Ctx (C function), 173
 nppiSubC_32sc_C1IRSfs_Ctx (C function), 172
 nppiSubC_32sc_C1RSfs_Ctx (C function), 171
 nppiSubC_32sc_C3IRSfs_Ctx (C function), 173
 nppiSubC_32sc_C3RSfs_Ctx (C function), 172
 nppiSubDeviceC_8u_AC4IRSfs_Ctx (C function), 151
 nppiSubDeviceC_8u_AC4RSfs_Ctx (C function), 150
 nppiSubDeviceC_8u_C1IRSfs_Ctx (C function), 147
 nppiSubDeviceC_8u_C1RSfs_Ctx (C function), 147
 nppiSubDeviceC_8u_C3IRSfs_Ctx (C function), 149
 nppiSubDeviceC_8u_C3RSfs_Ctx (C function), 148
 nppiSubDeviceC_8u_C4IRSfs_Ctx (C function), 152
 nppiSubDeviceC_8u_C4RSfs_Ctx (C function), 151
 nppiSubDeviceC_16f_C1IR_Ctx (C function), 175
 nppiSubDeviceC_16f_C1R_Ctx (C function), 174

nppiSubDeviceC_16f_C3IR_Ctx (C function), 176
 nppiSubDeviceC_16f_C3R_Ctx (C function), 176
 nppiSubDeviceC_16f_C4IR_Ctx (C function), 178
 nppiSubDeviceC_16f_C4R_Ctx (C function), 177
 nppiSubDeviceC_16s_AC4IRSfs_Ctx (C function), 164
 nppiSubDeviceC_16s_AC4RSfs_Ctx (C function), 163
 nppiSubDeviceC_16s_C1IRSfs_Ctx (C function), 160
 nppiSubDeviceC_16s_C1RSfs_Ctx (C function), 160
 nppiSubDeviceC_16s_C3IRSfs_Ctx (C function), 162
 nppiSubDeviceC_16s_C3RSfs_Ctx (C function), 161
 nppiSubDeviceC_16s_C4IRSfs_Ctx (C function), 165
 nppiSubDeviceC_16s_C4RSfs_Ctx (C function), 165
 nppiSubDeviceC_16u_AC4IRSfs_Ctx (C function), 157
 nppiSubDeviceC_16u_AC4RSfs_Ctx (C function), 156
 nppiSubDeviceC_16u_C1IRSfs_Ctx (C function), 154
 nppiSubDeviceC_16u_C1RSfs_Ctx (C function), 153
 nppiSubDeviceC_16u_C3IRSfs_Ctx (C function), 155
 nppiSubDeviceC_16u_C3RSfs_Ctx (C function), 155
 nppiSubDeviceC_16u_C4IRSfs_Ctx (C function), 159
 nppiSubDeviceC_16u_C4RSfs_Ctx (C function), 158
 nppiSubDeviceC_32f_AC4IR_Ctx (C function), 182
 nppiSubDeviceC_32f_AC4R_Ctx (C function), 181
 nppiSubDeviceC_32f_C1IR_Ctx (C function), 179
 nppiSubDeviceC_32f_C1R_Ctx (C function), 179
 nppiSubDeviceC_32f_C3IR_Ctx (C function), 181
 nppiSubDeviceC_32f_C3R_Ctx (C function), 180
 nppiSubDeviceC_32f_C4IR_Ctx (C function), 183
 nppiSubDeviceC_32f_C4R_Ctx (C function), 183
 nppiSubDeviceC_32s_C1IRSfs_Ctx (C function), 169
 nppiSubDeviceC_32s_C1RSfs_Ctx (C function), 169
 nppiSubDeviceC_32s_C3IRSfs_Ctx (C function), 171
 nppiSubDeviceC_32s_C3RSfs_Ctx (C function), 170
 nppiSum_8u64s_C1R_Ctx (C function), 1357
 nppiSum_8u64s_C4R_Ctx (C function), 1359
 nppiSum_8u_AC4R_Ctx (C function), 1358
 nppiSum_8u_C1R_Ctx (C function), 1357
 nppiSum_8u_C3R_Ctx (C function), 1358
 nppiSum_8u_C4R_Ctx (C function), 1359
 nppiSum_16s_AC4R_Ctx (C function), 1358
 nppiSum_16s_C1R_Ctx (C function), 1358
 nppiSum_16s_C3R_Ctx (C function), 1358
 nppiSum_16s_C4R_Ctx (C function), 1359
 nppiSum_16u_AC4R_Ctx (C function), 1358
 nppiSum_16u_C1R_Ctx (C function), 1357
 nppiSum_16u_C3R_Ctx (C function), 1358
 nppiSum_16u_C4R_Ctx (C function), 1359
 nppiSum_32f_AC4R_Ctx (C function), 1359
 nppiSum_32f_C1R_Ctx (C function), 1358
 nppiSum_32f_C3R_Ctx (C function), 1358
 nppiSum_32f_C4R_Ctx (C function), 1359
 nppiSumGetBufferHostSize_8u64s_C1R_Ctx (C function), 1360
 nppiSumGetBufferHostSize_8u64s_C4R_Ctx (C function), 1361
 nppiSumGetBufferHostSize_8u_AC4R_Ctx (C function), 1361
 nppiSumGetBufferHostSize_8u_C1R_Ctx (C function), 1360
 nppiSumGetBufferHostSize_8u_C3R_Ctx (C function), 1360
 nppiSumGetBufferHostSize_8u_C4R_Ctx (C function), 1361
 nppiSumGetBufferHostSize_16s_AC4R_Ctx (C function), 1361
 nppiSumGetBufferHostSize_16s_C1R_Ctx (C function), 1360
 nppiSumGetBufferHostSize_16s_C3R_Ctx (C function), 1361
 nppiSumGetBufferHostSize_16s_C4R_Ctx (C function), 1361
 nppiSumGetBufferHostSize_16u_AC4R_Ctx (C function), 1361
 nppiSumGetBufferHostSize_16u_C1R_Ctx (C function), 1360
 nppiSumGetBufferHostSize_16u_C3R_Ctx (C function), 1360

nppiSumGetBufferHostSize_16u_C4R_Ctx (C function), 1361	nppiSumWindowRow_16s32f_C3R_Ctx (C function), 930
nppiSumGetBufferHostSize_32f_AC4R_Ctx (C function), 1361	nppiSumWindowRow_16s32f_C4R_Ctx (C function), 931
nppiSumGetBufferHostSize_32f_C1R_Ctx (C function), 1360	nppiSumWindowRow_16u32f_C1R_Ctx (C function), 929
nppiSumGetBufferHostSize_32f_C3R_Ctx (C function), 1361	nppiSumWindowRow_16u32f_C3R_Ctx (C function), 930
nppiSumGetBufferHostSize_32f_C4R_Ctx (C function), 1362	nppiSumWindowRow_16u32f_C4R_Ctx (C function), 930
nppiSumWindowColumn_8u32f_C1R_Ctx (C function), 925	nppiSumWindowRowBorder_8u32f_C1R_Ctx (C function), 936
nppiSumWindowColumn_8u32f_C3R_Ctx (C function), 926	nppiSumWindowRowBorder_8u32f_C3R_Ctx (C function), 936
nppiSumWindowColumn_8u32f_C4R_Ctx (C function), 926	nppiSumWindowRowBorder_8u32f_C4R_Ctx (C function), 937
nppiSumWindowColumn_16s32f_C1R_Ctx (C function), 927	nppiSumWindowRowBorder_16s32f_C1R_Ctx (C function), 938
nppiSumWindowColumn_16s32f_C3R_Ctx (C function), 927	nppiSumWindowRowBorder_16s32f_C3R_Ctx (C function), 938
nppiSumWindowColumn_16s32f_C4R_Ctx (C function), 928	nppiSumWindowRowBorder_16s32f_C4R_Ctx (C function), 939
nppiSumWindowColumn_16u32f_C1R_Ctx (C function), 926	nppiSumWindowRowBorder_16u32f_C1R_Ctx (C function), 937
nppiSumWindowColumn_16u32f_C3R_Ctx (C function), 926	nppiSumWindowRowBorder_16u32f_C3R_Ctx (C function), 937
nppiSumWindowColumn_16u32f_C4R_Ctx (C function), 927	nppiSumWindowRowBorder_16u32f_C4R_Ctx (C function), 938
nppiSumWindowColumnBorder_8u32f_C1R_Ctx (C function), 932	nppiSwapChannels_8u_AC4R_Ctx (C function), 883
nppiSumWindowColumnBorder_8u32f_C3R_Ctx (C function), 933	nppiSwapChannels_8u_C3C4R_Ctx (C function), 882
nppiSumWindowColumnBorder_8u32f_C4R_Ctx (C function), 933	nppiSwapChannels_8u_C3IR_Ctx (C function), 880
nppiSumWindowColumnBorder_16s32f_C1R_Ctx (C function), 934	nppiSwapChannels_8u_C3R_Ctx (C function), 880
nppiSumWindowColumnBorder_16s32f_C3R_Ctx (C function), 935	nppiSwapChannels_8u_C4C3R_Ctx (C function), 881
nppiSumWindowColumnBorder_16s32f_C4R_Ctx (C function), 935	nppiSwapChannels_8u_C4IR_Ctx (C function), 882
nppiSumWindowColumnBorder_16u32f_C1R_Ctx (C function), 933	nppiSwapChannels_8u_C4R_Ctx (C function), 881
nppiSumWindowColumnBorder_16u32f_C3R_Ctx (C function), 934	nppiSwapChannels_16s_AC4R_Ctx (C function), 889
nppiSumWindowColumnBorder_16u32f_C4R_Ctx (C function), 934	nppiSwapChannels_16s_C3C4R_Ctx (C function), 888
nppiSumWindowRow_8u32f_C1R_Ctx (C function), 929	nppiSwapChannels_16s_C3IR_Ctx (C function), 887
nppiSumWindowRow_8u32f_C3R_Ctx (C function), 929	nppiSwapChannels_16s_C3R_Ctx (C function), 886
nppiSumWindowRow_8u32f_C4R_Ctx (C function), 929	nppiSwapChannels_16s_C4C3R_Ctx (C function), 887
nppiSumWindowRow_16s32f_C1R_Ctx (C function), 930	nppiSwapChannels_16s_C4IR_Ctx (C function), 888

nppiSwapChannels_16s_C4R_Ctx (C function), 888
 nppiSwapChannels_16u_AC4R_Ctx (C function), 886
 nppiSwapChannels_16u_C3C4R_Ctx (C function), 885
 nppiSwapChannels_16u_C3IR_Ctx (C function), 884
 nppiSwapChannels_16u_C3R_Ctx (C function), 883
 nppiSwapChannels_16u_C4C3R_Ctx (C function), 884
 nppiSwapChannels_16u_C4IR_Ctx (C function), 885
 nppiSwapChannels_16u_C4R_Ctx (C function), 884
 nppiSwapChannels_32f_AC4R_Ctx (C function), 895
 nppiSwapChannels_32f_C3C4R_Ctx (C function), 895
 nppiSwapChannels_32f_C3IR_Ctx (C function), 893
 nppiSwapChannels_32f_C3R_Ctx (C function), 893
 nppiSwapChannels_32f_C4C3R_Ctx (C function), 893
 nppiSwapChannels_32f_C4IR_Ctx (C function), 894
 nppiSwapChannels_32f_C4R_Ctx (C function), 894
 nppiSwapChannels_32s_AC4R_Ctx (C function), 892
 nppiSwapChannels_32s_C3C4R_Ctx (C function), 891
 nppiSwapChannels_32s_C3IR_Ctx (C function), 890
 nppiSwapChannels_32s_C3R_Ctx (C function), 889
 nppiSwapChannels_32s_C4C3R_Ctx (C function), 890
 nppiSwapChannels_32s_C4IR_Ctx (C function), 891
 nppiSwapChannels_32s_C4R_Ctx (C function), 891
 nppiThreshold_8u_AC4IR_Ctx (C function), 1663
 nppiThreshold_8u_AC4R_Ctx (C function), 1663
 nppiThreshold_8u_C1IR_Ctx (C function), 1660
 nppiThreshold_8u_C1R_Ctx (C function), 1659
 nppiThreshold_8u_C3IR_Ctx (C function), 1661
 nppiThreshold_8u_C3R_Ctx (C function), 1661
 nppiThreshold_16s_AC4IR_Ctx (C function), 1664
 nppiThreshold_16s_AC4R_Ctx (C function), 1663
 nppiThreshold_16s_C1IR_Ctx (C function), 1660
 nppiThreshold_16s_C1R_Ctx (C function), 1660
 nppiThreshold_16s_C3IR_Ctx (C function), 1662
 nppiThreshold_16s_C3R_Ctx (C function), 1662
 nppiThreshold_16u_AC4IR_Ctx (C function), 1663
 nppiThreshold_16u_AC4R_Ctx (C function), 1663
 nppiThreshold_16u_C1IR_Ctx (C function), 1660
 nppiThreshold_16u_C1R_Ctx (C function), 1660
 nppiThreshold_16u_C3IR_Ctx (C function), 1662
 nppiThreshold_16u_C3R_Ctx (C function), 1661
 nppiThreshold_32f_AC4IR_Ctx (C function), 1664
 nppiThreshold_32f_AC4R_Ctx (C function), 1664
 nppiThreshold_32f_C1IR_Ctx (C function), 1661
 nppiThreshold_32f_C1R_Ctx (C function), 1661
 nppiThreshold_32f_C3IR_Ctx (C function), 1662
 nppiThreshold_32f_C3R_Ctx (C function), 1662
 nppiThreshold_GT_8u_AC4IR_Ctx (C function), 1668
 nppiThreshold_GT_8u_AC4R_Ctx (C function), 1668
 nppiThreshold_GT_8u_C1IR_Ctx (C function), 1665
 nppiThreshold_GT_8u_C1R_Ctx (C function), 1665
 nppiThreshold_GT_8u_C3IR_Ctx (C function), 1667
 nppiThreshold_GT_8u_C3R_Ctx (C function), 1666
 nppiThreshold_GT_16s_AC4IR_Ctx (C function), 1669
 nppiThreshold_GT_16s_AC4R_Ctx (C function), 1669
 nppiThreshold_GT_16s_C1IR_Ctx (C function), 1666

nppiThreshold_GT_16s_C1R_Ctx (C function), 1666	nppiThreshold_GTVal_16u_AC4IR_Ctx (C function), 1686
nppiThreshold_GT_16s_C3IR_Ctx (C function), 1667	nppiThreshold_GTVal_16u_AC4R_Ctx (C function), 1685
nppiThreshold_GT_16s_C3R_Ctx (C function), 1667	nppiThreshold_GTVal_16u_C1IR_Ctx (C function), 1682
nppiThreshold_GT_16u_AC4IR_Ctx (C function), 1668	nppiThreshold_GTVal_16u_C1R_Ctx (C function), 1682
nppiThreshold_GT_16u_AC4R_Ctx (C function), 1668	nppiThreshold_GTVal_16u_C3IR_Ctx (C function), 1684
nppiThreshold_GT_16u_C1IR_Ctx (C function), 1665	nppiThreshold_GTVal_16u_C3R_Ctx (C function), 1684
nppiThreshold_GT_16u_C1R_Ctx (C function), 1665	nppiThreshold_GTVal_32f_AC4IR_Ctx (C function), 1686
nppiThreshold_GT_16u_C3IR_Ctx (C function), 1667	nppiThreshold_GTVal_32f_AC4R_Ctx (C function), 1686
nppiThreshold_GT_16u_C3R_Ctx (C function), 1667	nppiThreshold_GTVal_32f_C1IR_Ctx (C function), 1683
nppiThreshold_GT_32f_AC4IR_Ctx (C function), 1669	nppiThreshold_GTVal_32f_C1R_Ctx (C function), 1683
nppiThreshold_GT_32f_AC4R_Ctx (C function), 1669	nppiThreshold_GTVal_32f_C3IR_Ctx (C function), 1685
nppiThreshold_GT_32f_C1IR_Ctx (C function), 1666	nppiThreshold_GTVal_32f_C3R_Ctx (C function), 1685
nppiThreshold_GT_32f_C1R_Ctx (C function), 1666	nppiThreshold_LT_8u_AC4IR_Ctx (C function), 1673
nppiThreshold_GT_32f_C3IR_Ctx (C function), 1668	nppiThreshold_LT_8u_AC4R_Ctx (C function), 1673
nppiThreshold_GT_32f_C3R_Ctx (C function), 1667	nppiThreshold_LT_8u_C1IR_Ctx (C function), 1670
nppiThreshold_GTVal_8u_AC4IR_Ctx (C function), 1685	nppiThreshold_LT_8u_C1R_Ctx (C function), 1670
nppiThreshold_GTVal_8u_AC4R_Ctx (C function), 1685	nppiThreshold_LT_8u_C3IR_Ctx (C function), 1672
nppiThreshold_GTVal_8u_C1IR_Ctx (C function), 1682	nppiThreshold_LT_8u_C3R_Ctx (C function), 1672
nppiThreshold_GTVal_8u_C1R_Ctx (C function), 1682	nppiThreshold_LT_16s_AC4IR_Ctx (C function), 1674
nppiThreshold_GTVal_8u_C3IR_Ctx (C function), 1684	nppiThreshold_LT_16s_AC4R_Ctx (C function), 1674
nppiThreshold_GTVal_8u_C3R_Ctx (C function), 1683	nppiThreshold_LT_16s_C1IR_Ctx (C function), 1671
nppiThreshold_GTVal_16s_AC4IR_Ctx (C function), 1686	nppiThreshold_LT_16s_C1R_Ctx (C function), 1671
nppiThreshold_GTVal_16s_AC4R_Ctx (C function), 1686	nppiThreshold_LT_16s_C3IR_Ctx (C function), 1673
nppiThreshold_GTVal_16s_C1IR_Ctx (C function), 1683	nppiThreshold_LT_16s_C3R_Ctx (C function), 1672
nppiThreshold_GTVal_16s_C1R_Ctx (C function), 1683	nppiThreshold_LT_16u_AC4IR_Ctx (C function), 1674
nppiThreshold_GTVal_16s_C3IR_Ctx (C function), 1684	nppiThreshold_LT_16u_AC4R_Ctx (C function), 1674
nppiThreshold_GTVal_16s_C3R_Ctx (C function), 1684	nppiThreshold_LT_16u_C1IR_Ctx (C function), 1671

nppiThreshold_LT_16u_C1R_Ctx (C function), 1671
 nppiThreshold_LT_16u_C3IR_Ctx (C function), 1672
 nppiThreshold_LT_16u_C3R_Ctx (C function), 1672
 nppiThreshold_LT_32f_AC4IR_Ctx (C function), 1674
 nppiThreshold_LT_32f_AC4R_Ctx (C function), 1674
 nppiThreshold_LT_32f_C1IR_Ctx (C function), 1672
 nppiThreshold_LT_32f_C1R_Ctx (C function), 1671
 nppiThreshold_LT_32f_C3IR_Ctx (C function), 1673
 nppiThreshold_LT_32f_C3R_Ctx (C function), 1673
 nppiThreshold_LTVa1_8u_AC4IR_Ctx (C function), 1691
 nppiThreshold_LTVa1_8u_AC4R_Ctx (C function), 1691
 nppiThreshold_LTVa1_8u_C1IR_Ctx (C function), 1688
 nppiThreshold_LTVa1_8u_C1R_Ctx (C function), 1687
 nppiThreshold_LTVa1_8u_C3IR_Ctx (C function), 1689
 nppiThreshold_LTVa1_8u_C3R_Ctx (C function), 1689
 nppiThreshold_LTVa1_16s_AC4IR_Ctx (C function), 1692
 nppiThreshold_LTVa1_16s_AC4R_Ctx (C function), 1691
 nppiThreshold_LTVa1_16s_C1IR_Ctx (C function), 1688
 nppiThreshold_LTVa1_16s_C1R_Ctx (C function), 1688
 nppiThreshold_LTVa1_16s_C3IR_Ctx (C function), 1690
 nppiThreshold_LTVa1_16s_C3R_Ctx (C function), 1690
 nppiThreshold_LTVa1_16u_AC4IR_Ctx (C function), 1691
 nppiThreshold_LTVa1_16u_AC4R_Ctx (C function), 1691
 nppiThreshold_LTVa1_16u_C1IR_Ctx (C function), 1688
 nppiThreshold_LTVa1_16u_C1R_Ctx (C function), 1688
 nppiThreshold_LTVa1_16u_C3IR_Ctx (C function), 1690
 nppiThreshold_LTVa1_16u_C3R_Ctx (C function), 1689
 nppiThreshold_LTVa1_32f_AC4IR_Ctx (C function), 1692
 nppiThreshold_LTVa1_32f_AC4R_Ctx (C function), 1692
 nppiThreshold_LTVa1_32f_C1IR_Ctx (C function), 1689
 nppiThreshold_LTVa1_32f_C1R_Ctx (C function), 1689
 nppiThreshold_LTVa1_32f_C3IR_Ctx (C function), 1690
 nppiThreshold_LTVa1_32f_C3R_Ctx (C function), 1690
 nppiThreshold_LTVa1GTVa1_8u_AC4IR_Ctx (C function), 1699
 nppiThreshold_LTVa1GTVa1_8u_AC4R_Ctx (C function), 1699
 nppiThreshold_LTVa1GTVa1_8u_C1IR_Ctx (C function), 1695
 nppiThreshold_LTVa1GTVa1_8u_C1R_Ctx (C function), 1695
 nppiThreshold_LTVa1GTVa1_8u_C3IR_Ctx (C function), 1697
 nppiThreshold_LTVa1GTVa1_8u_C3R_Ctx (C function), 1697
 nppiThreshold_LTVa1GTVa1_16s_AC4IR_Ctx (C function), 1700
 nppiThreshold_LTVa1GTVa1_16s_AC4R_Ctx (C function), 1700
 nppiThreshold_LTVa1GTVa1_16s_C1IR_Ctx (C function), 1696
 nppiThreshold_LTVa1GTVa1_16s_C1R_Ctx (C function), 1696
 nppiThreshold_LTVa1GTVa1_16s_C3IR_Ctx (C function), 1698
 nppiThreshold_LTVa1GTVa1_16s_C3R_Ctx (C function), 1698
 nppiThreshold_LTVa1GTVa1_16u_AC4IR_Ctx (C function), 1700
 nppiThreshold_LTVa1GTVa1_16u_AC4R_Ctx (C function), 1699
 nppiThreshold_LTVa1GTVa1_16u_C1IR_Ctx (C function), 1695
 nppiThreshold_LTVa1GTVa1_16u_C1R_Ctx (C function), 1695
 nppiThreshold_LTVa1GTVa1_16u_C3IR_Ctx (C function), 1697
 nppiThreshold_LTVa1GTVa1_16u_C3R_Ctx (C function), 1697
 nppiThreshold_LTVa1GTVa1_32f_AC4IR_Ctx (C function), 1701
 nppiThreshold_LTVa1GTVa1_32f_AC4R_Ctx (C function), 1701
 nppiThreshold_LTVa1GTVa1_32f_C1IR_Ctx (C function), 1696

nppiThreshold_LTValGTVal_32f_C1R_Ctx (C function), 1696
 nppiThreshold_LTValGTVal_32f_C3IR_Ctx (C function), 1699
 nppiThreshold_LTValGTVal_32f_C3R_Ctx (C function), 1698
 nppiThreshold_Val_8u_AC4IR_Ctx (C function), 1679
 nppiThreshold_Val_8u_AC4R_Ctx (C function), 1679
 nppiThreshold_Val_8u_C1IR_Ctx (C function), 1676
 nppiThreshold_Val_8u_C1R_Ctx (C function), 1676
 nppiThreshold_Val_8u_C3IR_Ctx (C function), 1678
 nppiThreshold_Val_8u_C3R_Ctx (C function), 1677
 nppiThreshold_Val_16s_AC4IR_Ctx (C function), 1680
 nppiThreshold_Val_16s_AC4R_Ctx (C function), 1680
 nppiThreshold_Val_16s_C1IR_Ctx (C function), 1677
 nppiThreshold_Val_16s_C1R_Ctx (C function), 1676
 nppiThreshold_Val_16s_C3IR_Ctx (C function), 1678
 nppiThreshold_Val_16s_C3R_Ctx (C function), 1678
 nppiThreshold_Val_16u_AC4IR_Ctx (C function), 1680
 nppiThreshold_Val_16u_AC4R_Ctx (C function), 1680
 nppiThreshold_Val_16u_C1IR_Ctx (C function), 1676
 nppiThreshold_Val_16u_C1R_Ctx (C function), 1676
 nppiThreshold_Val_16u_C3IR_Ctx (C function), 1678
 nppiThreshold_Val_16u_C3R_Ctx (C function), 1678
 nppiThreshold_Val_32f_AC4IR_Ctx (C function), 1681
 nppiThreshold_Val_32f_AC4R_Ctx (C function), 1680
 nppiThreshold_Val_32f_C1IR_Ctx (C function), 1677
 nppiThreshold_Val_32f_C1R_Ctx (C function), 1677
 nppiThreshold_Val_32f_C3IR_Ctx (C function), 1679
 nppiThreshold_Val_32f_C3R_Ctx (C function), 1679
 nppiTranspose_8u_C1R_Ctx (C function), 878
 nppiTranspose_8u_C3R_Ctx (C function), 878
 nppiTranspose_8u_C4R_Ctx (C function), 878
 nppiTranspose_16s_C1R_Ctx (C function), 879
 nppiTranspose_16s_C3R_Ctx (C function), 879
 nppiTranspose_16s_C4R_Ctx (C function), 879
 nppiTranspose_16u_C1R_Ctx (C function), 878
 nppiTranspose_16u_C3R_Ctx (C function), 879
 nppiTranspose_16u_C4R_Ctx (C function), 879
 nppiTranspose_32f_C1R_Ctx (C function), 879
 nppiTranspose_32f_C3R_Ctx (C function), 880
 nppiTranspose_32f_C4R_Ctx (C function), 880
 nppiTranspose_32s_C1R_Ctx (C function), 879
 nppiTranspose_32s_C3R_Ctx (C function), 879
 nppiTranspose_32s_C4R_Ctx (C function), 879
 nppiUYVP_10u_ToI444_8u_C3P3R_Ctx (C function), 532
 nppiUYVP_10u_ToNV16_8u_C3P2R_Ctx (C function), 532
 nppiUYVP_10u_ToP208_8u_C3C3R_Ctx (C function), 531
 nppiUYVP_10u_ToRGB_8u_C3AC4R_Ctx (C function), 530
 nppiUYVP_10u_ToRGB_8u_C3C3R_Ctx (C function), 530
 nppiUYVP_10u_ToRGB_8u_C3P3R_Ctx (C function), 530
 nppiUYVP_10u_ToRGB_16f_C3C3R_Ctx (C function), 531
 nppiUYVP_10u_ToRGB_16f_C3P3R_Ctx (C function), 531
 nppiUYVP_10u_ToRGB_16u_C3AC4R_Ctx (C function), 531
 nppiUYVP_10u_ToRGB_16u_C3C3R_Ctx (C function), 530
 nppiUYVP_10u_ToRGB_16u_C3P3R_Ctx (C function), 531
 nppiUYVP_10u_ToRGBA_16f_C3AC4R_Ctx (C function), 531
 nppiUYVP_10u_ToV210_10u_C3C3R_Ctx (C function), 532
 nppiUYVP_10u_ToY410_10u_C3C3R_Ctx (C function), 532
 nppiUYVP_10u_ToYUV444_8u_C3C3R_Ctx (C function), 532
 nppiUYVY16_16u_ToRGB_16u_C3AC4R_Ctx (C function), 534
 nppiUYVY16_16u_ToRGB_16u_C3C3R_Ctx (C function), 534
 nppiUYVY16_16u_ToRGB_16u_C3P3R_Ctx (C function), 534
 nppiUYVY_8u_ToRGB_8u_C3AC4R_Ctx (C function), 534
 nppiUYVY_8u_ToRGB_8u_C3C3R_Ctx (C function), 534

tion), 534

nppiUYVY_8u_ToRGB_8u_C3P3R_Ctx (C function), 534

nppiV210_10u_ToUYVP_10u_C3C3R_Ctx (C function), 532

nppiValidNormLevelGetBufferHostSize_8s32fpp1WarpAffine_8u_C1R_Ctx (C function), 1270

nppiValidNormLevelGetBufferHostSize_8s32fpp1WarpAffine_8u_C3R_Ctx (C function), 1269

nppiValidNormLevelGetBufferHostSize_8s32fpp1WarpAffine_8u_C4R_Ctx (C function), 1269

nppiValidNormLevelGetBufferHostSize_8s32fpp3WarpAffine_8u_P3R_Ctx (C function), 1270

nppiValidNormLevelGetBufferHostSize_8s32fpp3WarpAffine_8u_P4R_Ctx (C function), 1270

nppiValidNormLevelGetBufferHostSize_8s32fpp4WarpAffine_16f_C1R_Ctx (C function), 1272

nppiValidNormLevelGetBufferHostSize_8u32fpp1WarpAffine_16f_C3R_Ctx (C function), 1272

nppiValidNormLevelGetBufferHostSize_8u32fpp1WarpAffine_16f_C4R_Ctx (C function), 1272

nppiValidNormLevelGetBufferHostSize_8u32fpp3WarpAffine_16u_AC4R_Ctx (C function), 1271

nppiValidNormLevelGetBufferHostSize_8u32fpp4WarpAffine_16u_C1R_Ctx (C function), 1270

nppiValidNormLevelGetBufferHostSize_8u32fpp1WarpAffine_16u_C3R_Ctx (C function), 1270

nppiValidNormLevelGetBufferHostSize_8u32fpp1WarpAffine_16u_C4R_Ctx (C function), 1270

nppiValidNormLevelGetBufferHostSize_8u32fpp3WarpAffine_16u_P3R_Ctx (C function), 1271

nppiValidNormLevelGetBufferHostSize_8u32fpp4WarpAffine_16u_P4R_Ctx (C function), 1271

nppiValidNormLevelGetBufferHostSize_16u32fpp1WarpAffine_32f_AC4R_Ctx (C function), 1273

nppiValidNormLevelGetBufferHostSize_16u32fpp1WarpAffine_32f_C1R_Ctx (C function), 1272

nppiValidNormLevelGetBufferHostSize_16u32fpp1WarpAffine_32f_C3R_Ctx (C function), 1273

nppiValidNormLevelGetBufferHostSize_16u32fpp1WarpAffine_32f_C4R_Ctx (C function), 1273

nppiValidNormLevelGetBufferHostSize_32fpp1WarpAffine_32f_P3R_Ctx (C function), 1273

nppiValidNormLevelGetBufferHostSize_32fpp1WarpAffine_32f_P4R_Ctx (C function), 1273

nppiValidNormLevelGetBufferHostSize_32fpp1WarpAffine_32s_AC4R_Ctx (C function), 1272

nppiValidNormLevelGetBufferHostSize_32fpp1WarpAffine_32s_C1R_Ctx (C function), 1271

nppiValidNormLevelGetBufferHostSize_64fpp1WarpAffine_32s_C3R_Ctx (C function), 1271

nppiValidNormLevelGetBufferHostSize_64fpp1WarpAffine_32s_C4R_Ctx (C function), 1271

nppiValidNormLevelGetBufferHostSize_64fpp1WarpAffine_32s_P3R_Ctx (C function), 1272

nppiValidNormLevelGetBufferHostSize_64fpp1WarpAffine_32s_P4R_Ctx (C function), 1272

nppiWarpAffine_8u_AC4R_Ctx (C function), nppiWarpAffine_64f_AC4R_Ctx (C function),

[1274](#)
 nppiWarpAffine_64f_C1R_Ctx (C function), [1273](#)
 nppiWarpAffine_64f_C3R_Ctx (C function), [1274](#)
 nppiWarpAffine_64f_C4R_Ctx (C function), [1274](#)
 nppiWarpAffine_64f_P3R_Ctx (C function), [1274](#)
 nppiWarpAffine_64f_P4R_Ctx (C function), [1274](#)
 nppiWarpAffineBack_8u_AC4R_Ctx (C function), [1280](#)
 nppiWarpAffineBack_8u_C1R_Ctx (C function), [1280](#)
 nppiWarpAffineBack_8u_C3R_Ctx (C function), [1280](#)
 nppiWarpAffineBack_8u_C4R_Ctx (C function), [1280](#)
 nppiWarpAffineBack_8u_P3R_Ctx (C function), [1281](#)
 nppiWarpAffineBack_8u_P4R_Ctx (C function), [1281](#)
 nppiWarpAffineBack_16u_AC4R_Ctx (C function), [1281](#)
 nppiWarpAffineBack_16u_C1R_Ctx (C function), [1281](#)
 nppiWarpAffineBack_16u_C3R_Ctx (C function), [1281](#)
 nppiWarpAffineBack_16u_C4R_Ctx (C function), [1281](#)
 nppiWarpAffineBack_16u_P3R_Ctx (C function), [1282](#)
 nppiWarpAffineBack_16u_P4R_Ctx (C function), [1282](#)
 nppiWarpAffineBack_32f_AC4R_Ctx (C function), [1284](#)
 nppiWarpAffineBack_32f_C1R_Ctx (C function), [1283](#)
 nppiWarpAffineBack_32f_C3R_Ctx (C function), [1283](#)
 nppiWarpAffineBack_32f_C4R_Ctx (C function), [1283](#)
 nppiWarpAffineBack_32f_P3R_Ctx (C function), [1284](#)
 nppiWarpAffineBack_32f_P4R_Ctx (C function), [1284](#)
 nppiWarpAffineBack_32s_AC4R_Ctx (C function), [1283](#)
 nppiWarpAffineBack_32s_C1R_Ctx (C function), [1282](#)
 nppiWarpAffineBack_32s_C3R_Ctx (C function), [1282](#)
 nppiWarpAffineBack_32s_C4R_Ctx (C function), [1282](#)
 nppiWarpAffineBack_32s_P3R_Ctx (C function), [1283](#)
 nppiWarpAffineBack_32s_P4R_Ctx (C function), [1283](#)
 nppiWarpAffineBatch_8u_AC4R_Ctx (C function), [1277](#)
 nppiWarpAffineBatch_8u_C1R_Ctx (C function), [1276](#)
 nppiWarpAffineBatch_8u_C3R_Ctx (C function), [1276](#)
 nppiWarpAffineBatch_8u_C4R_Ctx (C function), [1276](#)
 nppiWarpAffineBatch_16f_C1R_Ctx (C function), [1277](#)
 nppiWarpAffineBatch_16f_C3R_Ctx (C function), [1277](#)
 nppiWarpAffineBatch_16f_C4R_Ctx (C function), [1277](#)
 nppiWarpAffineBatch_32f_AC4R_Ctx (C function), [1278](#)
 nppiWarpAffineBatch_32f_C1R_Ctx (C function), [1277](#)
 nppiWarpAffineBatch_32f_C3R_Ctx (C function), [1278](#)
 nppiWarpAffineBatch_32f_C4R_Ctx (C function), [1278](#)
 NppiWarpAffineBatchCXR (C struct), [50](#)
 NppiWarpAffineBatchCXR.aTransformedCoeffs (C var), [50](#)
 NppiWarpAffineBatchCXR.nDstStep (C var), [50](#)
 NppiWarpAffineBatchCXR.nSrcStep (C var), [50](#)
 NppiWarpAffineBatchCXR.pCoeffs (C var), [50](#)
 NppiWarpAffineBatchCXR.pDst (C var), [50](#)
 NppiWarpAffineBatchCXR.pSrc (C var), [50](#)
 nppiWarpAffineBatchInit_Ctx (C function), [1276](#)
 nppiWarpAffineQuad_8u_AC4R_Ctx (C function), [1286](#)
 nppiWarpAffineQuad_8u_C1R_Ctx (C function), [1286](#)
 nppiWarpAffineQuad_8u_C3R_Ctx (C function), [1286](#)
 nppiWarpAffineQuad_8u_C4R_Ctx (C function), [1286](#)
 nppiWarpAffineQuad_8u_P3R_Ctx (C function), [1287](#)
 nppiWarpAffineQuad_8u_P4R_Ctx (C function), [1287](#)
 nppiWarpAffineQuad_16u_AC4R_Ctx (C function), [1288](#)
 nppiWarpAffineQuad_16u_C1R_Ctx (C function), [1288](#)

tion), 1287
 nppiWarpAffineQuad_16u_C3R_Ctx (C func-
 tion), 1287
 nppiWarpAffineQuad_16u_C4R_Ctx (C func-
 tion), 1287
 nppiWarpAffineQuad_16u_P3R_Ctx (C func-
 tion), 1288
 nppiWarpAffineQuad_16u_P4R_Ctx (C func-
 tion), 1288
 nppiWarpAffineQuad_32f_AC4R_Ctx (C func-
 tion), 1290
 nppiWarpAffineQuad_32f_C1R_Ctx (C func-
 tion), 1289
 nppiWarpAffineQuad_32f_C3R_Ctx (C func-
 tion), 1290
 nppiWarpAffineQuad_32f_C4R_Ctx (C func-
 tion), 1290
 nppiWarpAffineQuad_32f_P3R_Ctx (C func-
 tion), 1290
 nppiWarpAffineQuad_32f_P4R_Ctx (C func-
 tion), 1290
 nppiWarpAffineQuad_32s_AC4R_Ctx (C func-
 tion), 1289
 nppiWarpAffineQuad_32s_C1R_Ctx (C func-
 tion), 1288
 nppiWarpAffineQuad_32s_C3R_Ctx (C func-
 tion), 1288
 nppiWarpAffineQuad_32s_C4R_Ctx (C func-
 tion), 1289
 nppiWarpAffineQuad_32s_P3R_Ctx (C func-
 tion), 1289
 nppiWarpAffineQuad_32s_P4R_Ctx (C func-
 tion), 1289
 nppiWarpPerspective_8u_AC4R_Ctx (C func-
 tion), 1295
 nppiWarpPerspective_8u_C1R_Ctx (C func-
 tion), 1294
 nppiWarpPerspective_8u_C3R_Ctx (C func-
 tion), 1294
 nppiWarpPerspective_8u_C4R_Ctx (C func-
 tion), 1295
 nppiWarpPerspective_8u_P3R_Ctx (C func-
 tion), 1295
 nppiWarpPerspective_8u_P4R_Ctx (C func-
 tion), 1295
 nppiWarpPerspective_16f_C1R_Ctx (C func-
 tion), 1297
 nppiWarpPerspective_16f_C3R_Ctx (C func-
 tion), 1298
 nppiWarpPerspective_16f_C4R_Ctx (C func-
 tion), 1298
 nppiWarpPerspective_16u_AC4R_Ctx (C func-
 tion), 1296
 nppiWarpPerspective_16u_C1R_Ctx (C func-
 tion), 1295
 nppiWarpPerspective_16u_C3R_Ctx (C func-
 tion), 1296
 nppiWarpPerspective_16u_C4R_Ctx (C func-
 tion), 1296
 nppiWarpPerspective_16u_P3R_Ctx (C func-
 tion), 1296
 nppiWarpPerspective_16u_P4R_Ctx (C func-
 tion), 1296
 nppiWarpPerspective_32f_AC4R_Ctx (C func-
 tion), 1299
 nppiWarpPerspective_32f_C1R_Ctx (C func-
 tion), 1298
 nppiWarpPerspective_32f_C3R_Ctx (C func-
 tion), 1298
 nppiWarpPerspective_32f_C4R_Ctx (C func-
 tion), 1298
 nppiWarpPerspective_32f_P3R_Ctx (C func-
 tion), 1299
 nppiWarpPerspective_32f_P4R_Ctx (C func-
 tion), 1299
 nppiWarpPerspective_32s_AC4R_Ctx (C func-
 tion), 1297
 nppiWarpPerspective_32s_C1R_Ctx (C func-
 tion), 1296
 nppiWarpPerspective_32s_C3R_Ctx (C func-
 tion), 1297
 nppiWarpPerspective_32s_C4R_Ctx (C func-
 tion), 1297
 nppiWarpPerspective_32s_P3R_Ctx (C func-
 tion), 1297
 nppiWarpPerspective_32s_P4R_Ctx (C func-
 tion), 1297
 nppiWarpPerspectiveBack_8u_AC4R_Ctx (C
 function), 1305
 nppiWarpPerspectiveBack_8u_C1R_Ctx (C
 function), 1305
 nppiWarpPerspectiveBack_8u_C3R_Ctx (C
 function), 1305
 nppiWarpPerspectiveBack_8u_C4R_Ctx (C
 function), 1305
 nppiWarpPerspectiveBack_8u_P3R_Ctx (C
 function), 1306
 nppiWarpPerspectiveBack_8u_P4R_Ctx (C
 function), 1306
 nppiWarpPerspectiveBack_16u_AC4R_Ctx (C
 function), 1306
 nppiWarpPerspectiveBack_16u_C1R_Ctx (C
 function), 1306
 nppiWarpPerspectiveBack_16u_C3R_Ctx (C
 function), 1306
 nppiWarpPerspectiveBack_16u_C4R_Ctx (C
 function), 1306
 nppiWarpPerspectiveBack_16u_P3R_Ctx (C

function), 1307
 nppiWarpPerspectiveBack_16u_P4R_Ctx (C *function*), 1307
 nppiWarpPerspectiveBack_32f_AC4R_Ctx (C *function*), 1309
 nppiWarpPerspectiveBack_32f_C1R_Ctx (C *function*), 1308
 nppiWarpPerspectiveBack_32f_C3R_Ctx (C *function*), 1308
 nppiWarpPerspectiveBack_32f_C4R_Ctx (C *function*), 1308
 nppiWarpPerspectiveBack_32f_P3R_Ctx (C *function*), 1309
 nppiWarpPerspectiveBack_32f_P4R_Ctx (C *function*), 1309
 nppiWarpPerspectiveBack_32s_AC4R_Ctx (C *function*), 1308
 nppiWarpPerspectiveBack_32s_C1R_Ctx (C *function*), 1307
 nppiWarpPerspectiveBack_32s_C3R_Ctx (C *function*), 1307
 nppiWarpPerspectiveBack_32s_C4R_Ctx (C *function*), 1307
 nppiWarpPerspectiveBack_32s_P3R_Ctx (C *function*), 1308
 nppiWarpPerspectiveBack_32s_P4R_Ctx (C *function*), 1308
 nppiWarpPerspectiveBatch_8u_AC4R_Ctx (C *function*), 1301
 nppiWarpPerspectiveBatch_8u_C1R_Ctx (C *function*), 1301
 nppiWarpPerspectiveBatch_8u_C3R_Ctx (C *function*), 1301
 nppiWarpPerspectiveBatch_8u_C4R_Ctx (C *function*), 1301
 nppiWarpPerspectiveBatch_16f_C1R_Ctx (C *function*), 1301
 nppiWarpPerspectiveBatch_16f_C3R_Ctx (C *function*), 1302
 nppiWarpPerspectiveBatch_16f_C4R_Ctx (C *function*), 1302
 nppiWarpPerspectiveBatch_32f_AC4R_Ctx (C *function*), 1302
 nppiWarpPerspectiveBatch_32f_C1R_Ctx (C *function*), 1302
 nppiWarpPerspectiveBatch_32f_C3R_Ctx (C *function*), 1302
 nppiWarpPerspectiveBatch_32f_C4R_Ctx (C *function*), 1302
 NppiWarpPerspectiveBatchCXR (C *struct*), 50
 NppiWarpPerspectiveBatchCXR.aTransformed (C *var*), 51
 NppiWarpPerspectiveBatchCXR.nDstStep (C *var*), 51
 NppiWarpPerspectiveBatchCXR.nSrcStep (C *var*), 51
 NppiWarpPerspectiveBatchCXR.pCoeffs (C *var*), 51
 NppiWarpPerspectiveBatchCXR.pDst (C *var*), 51
 NppiWarpPerspectiveBatchCXR.pSrc (C *var*), 51
 nppiWarpPerspectiveBatchInit_Ctx (C *function*), 1300
 nppiWarpPerspectiveQuad_8u_AC4R_Ctx (C *function*), 1312
 nppiWarpPerspectiveQuad_8u_C1R_Ctx (C *function*), 1311
 nppiWarpPerspectiveQuad_8u_C3R_Ctx (C *function*), 1311
 nppiWarpPerspectiveQuad_8u_C4R_Ctx (C *function*), 1311
 nppiWarpPerspectiveQuad_8u_P3R_Ctx (C *function*), 1312
 nppiWarpPerspectiveQuad_8u_P4R_Ctx (C *function*), 1312
 nppiWarpPerspectiveQuad_16u_AC4R_Ctx (C *function*), 1313
 nppiWarpPerspectiveQuad_16u_C1R_Ctx (C *function*), 1312
 nppiWarpPerspectiveQuad_16u_C3R_Ctx (C *function*), 1312
 nppiWarpPerspectiveQuad_16u_C4R_Ctx (C *function*), 1312
 nppiWarpPerspectiveQuad_16u_P3R_Ctx (C *function*), 1313
 nppiWarpPerspectiveQuad_16u_P4R_Ctx (C *function*), 1313
 nppiWarpPerspectiveQuad_32f_AC4R_Ctx (C *function*), 1315
 nppiWarpPerspectiveQuad_32f_C1R_Ctx (C *function*), 1314
 nppiWarpPerspectiveQuad_32f_C3R_Ctx (C *function*), 1315
 nppiWarpPerspectiveQuad_32f_C4R_Ctx (C *function*), 1315
 nppiWarpPerspectiveQuad_32f_P3R_Ctx (C *function*), 1315
 nppiWarpPerspectiveQuad_32f_P4R_Ctx (C *function*), 1315
 nppiWarpPerspectiveQuad_32s_AC4R_Ctx (C *function*), 1314
 nppiWarpPerspectiveQuad_32s_C1R_Ctx (C *function*), 1313
 nppiWarpPerspectiveQuad_32s_C3R_Ctx (C *function*), 1313
 nppiWarpPerspectiveQuad_32s_C4R_Ctx (C *function*), 1314

nppiWarpPerspectiveQuad_32s_P3R_Ctx (C function), 1314
 nppiWarpPerspectiveQuad_32s_P4R_Ctx (C function), 1314
 NppiWatershedSegmentBoundaryType (C enum), 47
 NppiWatershedSegmentBoundaryType.NPP_WATERSHED_SEGMENT_BOUNDARY_TYPE_BLACK (C enumerator), 47
 NppiWatershedSegmentBoundaryType.NPP_WATERSHED_SEGMENT_BOUNDARY_TYPE_CONTRAST (C enumerator), 47
 NppiWatershedSegmentBoundaryType.NPP_WATERSHED_SEGMENT_BOUNDARY_TYPE_NONE (C enumerator), 47
 NppiWatershedSegmentBoundaryType.NPP_WATERSHED_SEGMENT_BOUNDARY_TYPE_ORIGINAL (C enumerator), 47
 NppiWatershedSegmentBoundaryType.NPP_WATERSHED_SEGMENT_BOUNDARY_TYPE_WHITE (C enumerator), 47
 nppiWMSSSIM_8u_C1R_Ctx (C function), 1643
 nppiWMSSSIM_8u_C3R_Ctx (C function), 1644
 nppiWMSSSIM_8u_C4R_Ctx (C function), 1644
 nppiWMSSSIMBatch_8u_C1R_Advanced_Ctx (C function), 1656
 nppiWMSSSIMBatch_8u_C1R_Ctx (C function), 1650
 nppiWMSSSIMBatch_8u_C3R_Advanced_Ctx (C function), 1657
 nppiWMSSSIMBatch_8u_C3R_Ctx (C function), 1651
 nppiWMSSSIMBatch_8u_C4R_Advanced_Ctx (C function), 1657
 nppiWMSSSIMBatch_8u_C4R_Ctx (C function), 1651
 nppiWMSSSIMBatchGetBufferHostSize_8u_C1R_Ctx (C function), 1653
 nppiWMSSSIMBatchGetBufferHostSize_8u_C3R_Ctx (C function), 1653
 nppiWMSSSIMBatchGetBufferHostSize_8u_C4R_Ctx (C function), 1653
 nppiWMSSSIMGetBufferHostSize_8u_C1R_Ctx (C function), 1646
 nppiWMSSSIMGetBufferHostSize_8u_C3R_Ctx (C function), 1646
 nppiWMSSSIMGetBufferHostSize_8u_C4R_Ctx (C function), 1646
 nppiXor_8u_AC4IR_Ctx (C function), 481
 nppiXor_8u_AC4R_Ctx (C function), 481
 nppiXor_8u_C1IR_Ctx (C function), 480
 nppiXor_8u_C1R_Ctx (C function), 480
 nppiXor_8u_C3IR_Ctx (C function), 481
 nppiXor_8u_C3R_Ctx (C function), 480
 nppiXor_8u_C4IR_Ctx (C function), 482
 nppiXor_8u_C4R_Ctx (C function), 482
 nppiXor_16u_AC4IR_Ctx (C function), 485
 nppiXor_16u_AC4R_Ctx (C function), 484
 nppiXor_16u_C1IR_Ctx (C function), 483
 nppiXor_16u_C1R_Ctx (C function), 483
 nppiXor_16u_C3IR_Ctx (C function), 483
 nppiXor_16u_C3R_Ctx (C function), 485
 nppiXor_16u_C4IR_Ctx (C function), 485
 nppiXor_16u_C4R_Ctx (C function), 485
 nppiXor_32s_AC4IR_Ctx (C function), 488
 nppiXor_32s_C1IR_Ctx (C function), 486
 nppiXor_32s_C1R_Ctx (C function), 487
 nppiXor_32s_C3IR_Ctx (C function), 487
 nppiXor_32s_C3R_Ctx (C function), 486
 nppiXor_32s_C4IR_Ctx (C function), 488
 nppiXorC_8u_AC4IR_Ctx (C function), 431
 nppiXorC_8u_C1IR_Ctx (C function), 431
 nppiXorC_8u_C1R_Ctx (C function), 430
 nppiXorC_8u_C3IR_Ctx (C function), 431
 nppiXorC_8u_C3R_Ctx (C function), 430
 nppiXorC_8u_C4IR_Ctx (C function), 432
 nppiXorC_8u_C4R_Ctx (C function), 432
 nppiXorC_16u_AC4IR_Ctx (C function), 434
 nppiXorC_16u_AC4R_Ctx (C function), 434
 nppiXorC_16u_C1IR_Ctx (C function), 433
 nppiXorC_16u_C1R_Ctx (C function), 432
 nppiXorC_16u_C3IR_Ctx (C function), 433
 nppiXorC_16u_C3R_Ctx (C function), 433
 nppiXorC_16u_C4IR_Ctx (C function), 435
 nppiXorC_16u_C4R_Ctx (C function), 434
 nppiXorC_32s_AC4IR_Ctx (C function), 437
 nppiXorC_32s_AC4R_Ctx (C function), 437
 nppiXorC_32s_C1IR_Ctx (C function), 436
 nppiXorC_32s_C1R_Ctx (C function), 435
 nppiXorC_32s_C3IR_Ctx (C function), 436
 nppiXorC_32s_C3R_Ctx (C function), 436
 nppiXorC_32s_C4IR_Ctx (C function), 438
 nppiXorC_32s_C4R_Ctx (C function), 437
 nppiXYZToRGB_8u_AC4R_Ctx (C function), 769
 nppiXYZToRGB_8u_C3R_Ctx (C function), 769
 nppiY410_10u_ToUYVP_10u_C3C3R_Ctx (C function), 533
 nppiYCbCr411_8u_P2P3R_Ctx (C function), 688
 nppiYCbCr411_8u_P3P2R_Ctx (C function), 687
 nppiYCbCr411ToBGR_8u_P3C3R_Ctx (C function), 765
 nppiYCbCr411ToBGR_8u_P3C4R_Ctx (C function), 765
 nppiYCbCr411ToBGR_JPEG_8u_P3C3R_Ctx (C function), 804
 nppiYCbCr411ToBGR_JPEG_8u_P3R_Ctx (C function), 798
 nppiYCbCr411ToRGB_8u_P3C3R_Ctx (C function), 766

nppiYCbCr411ToRGB_8u_P3C4R_Ctx (C function), 766	nppiYCbCr420ToRGBBatch_8u_P3C3R_Ctx (C function), 752
nppiYCbCr411ToRGB_JPEG_8u_P3C3R_Ctx (C function), 802	nppiYCbCr420ToYCbCr411_8u_P2P3R_Ctx (C function), 668
nppiYCbCr411ToRGB_JPEG_8u_P3R_Ctx (C function), 796	nppiYCbCr420ToYCbCr411_8u_P3P2R_Ctx (C function), 667
nppiYCbCr411ToYCbCr420_8u_P2P3R_Ctx (C function), 692	nppiYCbCr420ToYCbCr422_8u_P2C2R_Ctx (C function), 683
nppiYCbCr411ToYCbCr420_8u_P3P2R_Ctx (C function), 692	nppiYCbCr420ToYCbCr422_8u_P2P3R_Ctx (C function), 683
nppiYCbCr411ToYCbCr420_8u_P3R_Ctx (C function), 691	nppiYCbCr420ToYCbCr422_8u_P3R_Ctx (C function), 683
nppiYCbCr411ToYCbCr422_8u_P2C2R_Ctx (C function), 690	nppiYCbCr420ToYCrCb420_8u_P2P3R_Ctx (C function), 685
nppiYCbCr411ToYCbCr422_8u_P2P3R_Ctx (C function), 689	nppiYCbCr422_8u_C2P3R_Ctx (C function), 669
nppiYCbCr411ToYCbCr422_8u_P3C2R_Ctx (C function), 689	nppiYCbCr422_8u_P3C2R_Ctx (C function), 669
nppiYCbCr411ToYCbCr422_8u_P3R_Ctx (C function), 688	nppiYCbCr422ToBGR_8u_C2C3R_Ctx (C function), 743
nppiYCbCr411ToYCrCb420_8u_P2P3R_Ctx (C function), 693	nppiYCbCr422ToBGR_8u_C2C4R_Ctx (C function), 743
nppiYCbCr411ToYCrCb422_8u_P3C2R_Ctx (C function), 691	nppiYCbCr422ToBGR_8u_P3C3R_Ctx (C function), 744
nppiYCbCr411ToYCrCb422_8u_P3R_Ctx (C function), 690	nppiYCbCr422ToBGR_JPEG_8u_P3C3R_Ctx (C function), 804
nppiYCbCr420_8u_P2P3R_Ctx (C function), 682	nppiYCbCr422ToBGR_JPEG_8u_P3R_Ctx (C function), 797
nppiYCbCr420_8u_P3P2R_Ctx (C function), 682	nppiYCbCr422ToBGRBatch_8u_P3C3R_Advanced_Ctx (C function), 745
nppiYCbCr420ToBGR_8u_P3C3R_Ctx (C function), 758	nppiYCbCr422ToBGRBatch_8u_P3C3R_Ctx (C function), 744
nppiYCbCr420ToBGR_8u_P3C4R_Ctx (C function), 759	nppiYCbCr422ToCbYCr422_8u_C2R_Ctx (C function), 671
nppiYCbCr420ToBGR_709CSC_8u_P3C3R_Ctx (C function), 761	nppiYCbCr422ToRGB_8u_C2C3R_Ctx (C function), 738
nppiYCbCr420ToBGR_709HDTV_8u_P3C4R_Ctx (C function), 761	nppiYCbCr422ToRGB_8u_C2P3R_Ctx (C function), 738
nppiYCbCr420ToBGR_JPEG_8u_P3C3R_Ctx (C function), 803	nppiYCbCr422ToRGB_8u_P3C3R_Ctx (C function), 739
nppiYCbCr420ToBGR_JPEG_8u_P3R_Ctx (C function), 797	nppiYCbCr422ToRGB_JPEG_8u_P3C3R_Ctx (C function), 802
nppiYCbCr420ToBGRBatch_8u_P3C3R_Advanced_Ctx (C function), 760	nppiYCbCr422ToRGB_JPEG_8u_P3R_Ctx (C function), 796
nppiYCbCr420ToBGRBatch_8u_P3C3R_Ctx (C function), 759	nppiYCbCr422ToRGBBatch_8u_P3C3R_Advanced_Ctx (C function), 740
nppiYCbCr420ToCbYCr422_8u_P2C2R_Ctx (C function), 684	nppiYCbCr422ToRGBBatch_8u_P3C3R_Ctx (C function), 740
nppiYCbCr420ToRGB_8u_P3C3R_Ctx (C function), 751	nppiYCbCr422ToYCbCr411_8u_C2P2R_Ctx (C function), 676
nppiYCbCr420ToRGB_JPEG_8u_P3C3R_Ctx (C function), 802	nppiYCbCr422ToYCbCr411_8u_C2P3R_Ctx (C function), 676
nppiYCbCr420ToRGB_JPEG_8u_P3R_Ctx (C function), 795	nppiYCbCr422ToYCbCr411_8u_P3P2R_Ctx (C function), 676
nppiYCbCr420ToRGBBatch_8u_P3C3R_Advanced_Ctx (C function), 753	nppiYCbCr422ToYCbCr411_8u_P3R_Ctx (C function), 675

npapiYCbCr422ToYCbCr420_8u_C2P2R_Ctx (C function), 673	npapiYCCCToCMYK_JPEG_601_8u_P4R_Ctx (C function), 778
npapiYCbCr422ToYCbCr420_8u_C2P3R_Ctx (C function), 673	npapiYCCToRGB_8u_AC4R_Ctx (C function), 777
npapiYCbCr422ToYCbCr420_8u_P3P2R_Ctx (C function), 672	npapiYCCToRGB_8u_C3R_Ctx (C function), 777
npapiYCbCr422ToYCbCr420_8u_P3R_Ctx (C function), 672	npapiYCrCb420ToCbYCr422_8u_P3C2R_Ctx (C function), 685
npapiYCbCr422ToYCrCb420_8u_C2P3R_Ctx (C function), 675	npapiYCrCb420ToRGB_8u_P3C4R_Ctx (C function), 754
npapiYCbCr422ToYCrCb422_8u_C2R_Ctx (C function), 670	npapiYCrCb420ToYCbCr411_8u_P3P2R_Ctx (C function), 687
npapiYCbCr422ToYCrCb422_8u_P3C2R_Ctx (C function), 670	npapiYCrCb420ToYCbCr420_8u_P3P2R_Ctx (C function), 686
npapiYCbCr444ToBGR_JPEG_8u_P3C3R_Ctx (C function), 804	npapiYCrCb420ToYCbCr422_8u_P3C2R_Ctx (C function), 674
npapiYCbCr444ToBGR_JPEG_8u_P3R_Ctx (C function), 798	npapiYCrCb420ToYCbCr422_8u_P3R_Ctx (C function), 674
npapiYCbCr444ToRGB_JPEG_8u_P3C3R_Ctx (C function), 803	npapiYCrCb422ToRGB_8u_C2C3R_Ctx (C function), 742
npapiYCbCr444ToRGB_JPEG_8u_P3R_Ctx (C function), 797	npapiYCrCb422ToRGB_8u_C2P3R_Ctx (C function), 742
npapiYCbCrToBGR_8u_P3C3R_Ctx (C function), 732	npapiYCrCb422ToYCbCr411_8u_C2P3R_Ctx (C function), 678
npapiYCbCrToBGR_8u_P3C4R_Ctx (C function), 733	npapiYCrCb422ToYCbCr420_8u_C2P3R_Ctx (C function), 678
npapiYCbCrToBGR_709CSC_8u_P3C3R_Ctx (C function), 736	npapiYCrCb422ToYCbCr422_8u_C2P3R_Ctx (C function), 677
npapiYCbCrToBGR_709CSC_8u_P3C4R_Ctx (C function), 736	npapiYUV420ToBGR_8u_P3C3R_Ctx (C function), 721
npapiYCbCrToBGRBatch_8u_C3R_Advanced_Ctx (C function), 735	npapiYUV420ToBGR_8u_P3C4R_Ctx (C function), 721
npapiYCbCrToBGRBatch_8u_C3R_Ctx (C function), 733	npapiYUV420ToBGRBatch_8u_P3C3R_Advanced_Ctx (C function), 722
npapiYCbCrToBGRBatch_8u_P3C3R_Advanced_Ctx (C function), 735	npapiYUV420ToBGRBatch_8u_P3C3R_Ctx (C function), 722
npapiYCbCrToBGRBatch_8u_P3C3R_Ctx (C function), 734	npapiYUV420ToNV12_8u_P3P2R_Ctx (C function), 536
npapiYCbCrToRGB_8u_AC4R_Ctx (C function), 728	npapiYUV420ToRGB_8u_ColorTwist32f_P3AC4R_Ctx (C function), 522
npapiYCbCrToRGB_8u_C3R_Ctx (C function), 728	npapiYUV420ToRGB_8u_ColorTwist32f_P3C3R_Ctx (C function), 520
npapiYCbCrToRGB_8u_P3C3R_Ctx (C function), 729	npapiYUV420ToRGB_8u_ColorTwist32f_P3C4R_Ctx (C function), 521
npapiYCbCrToRGB_8u_P3C4R_Ctx (C function), 729	npapiYUV420ToRGB_8u_ColorTwist32f_P3R_Ctx (C function), 519
npapiYCbCrToRGB_8u_P3R_Ctx (C function), 729	npapiYUV420ToRGB_8u_P3AC4R_Ctx (C function), 716
npapiYCbCrToRGBBatch_8u_C3R_Advanced_Ctx (C function), 731	npapiYUV420ToRGB_8u_P3C3R_Ctx (C function), 715
npapiYCbCrToRGBBatch_8u_C3R_Ctx (C function), 730	npapiYUV420ToRGB_8u_P3C4R_Ctx (C function), 715
npapiYCbCrToRGBBatch_8u_P3C3R_Advanced_Ctx (C function), 732	npapiYUV420ToRGB_8u_P3R_Ctx (C function), 715
npapiYCbCrToRGBBatch_8u_P3C3R_Ctx (C function), 730	npapiYUV420ToRGB_16u_ColorTwist32f_P3AC4R_Ctx (C function), 523

nppiYUV420ToRGB_16u_ColorTwist32f_P3C3R_Ctx (C function), 521
 nppiYUV420ToRGB_16u_ColorTwist32f_P3C4R_Ctx (C function), 522
 nppiYUV420ToRGB_16u_ColorTwist32f_P3R_Ctx (C function), 520
 nppiYUV420ToRGBBatch_8u_P3C3R_Advanced_Ctx (C function), 717
 nppiYUV420ToRGBBatch_8u_P3C3R_Ctx (C function), 716
 nppiYUV422ToBGRBatch_8u_P3C3R_Advanced_Ctx (C function), 713
 nppiYUV422ToBGRBatch_8u_P3C3R_Ctx (C function), 712
 nppiYUV422ToRGB_8u_C2C3R_Ctx (C function), 709
 nppiYUV422ToRGB_8u_ColorTwist32f_C2C3R_Ctx (C function), 523
 nppiYUV422ToRGB_8u_ColorTwist32f_P3AC4R_Ctx (C function), 526
 nppiYUV422ToRGB_8u_ColorTwist32f_P3C3R_Ctx (C function), 525
 nppiYUV422ToRGB_8u_ColorTwist32f_P3R_Ctx (C function), 524
 nppiYUV422ToRGB_8u_P3AC4R_Ctx (C function), 710
 nppiYUV422ToRGB_8u_P3C3R_Ctx (C function), 710
 nppiYUV422ToRGB_8u_P3R_Ctx (C function), 709
 nppiYUV422ToRGB_16u_ColorTwist32f_C2C3R_Ctx (C function), 524
 nppiYUV422ToRGB_16u_ColorTwist32f_P3AC4R_Ctx (C function), 526
 nppiYUV422ToRGB_16u_ColorTwist32f_P3C3R_Ctx (C function), 525
 nppiYUV422ToRGB_16u_ColorTwist32f_P3R_Ctx (C function), 525
 nppiYUV422ToRGBBatch_8u_P3C3R_Advanced_Ctx (C function), 711
 nppiYUV422ToRGBBatch_8u_P3C3R_Ctx (C function), 711
 nppiYUV444_8u_ToUYVP_10u_C3C3R_Ctx (C function), 532
 nppiYUVToBGR_8u_AC4R_Ctx (C function), 704
 nppiYUVToBGR_8u_C3R_Ctx (C function), 704
 nppiYUVToBGR_8u_P3C3R_Ctx (C function), 705
 nppiYUVToBGR_8u_P3R_Ctx (C function), 704
 nppiYUVToBGRBatch_8u_C3R_Advanced_Ctx (C function), 707
 nppiYUVToBGRBatch_8u_C3R_Ctx (C function), 705
 nppiYUVToBGRBatch_8u_P3C3R_Advanced_Ctx (C function), 707
 nppiYUVToBGRBatch_8u_P3C3R_Ctx (C function), 706
 nppiYUVToRGB_8u_AC4R_Ctx (C function), 699
 nppiYUVToRGB_8u_C3R_Ctx (C function), 699
 nppiYUVToRGB_8u_P3C3R_Ctx (C function), 700
 nppiYUVToRGB_8u_P3R_Ctx (C function), 700
 nppiYUVToRGBBatch_8u_C3R_Advanced_Ctx (C function), 702
 nppiYUVToRGBBatch_8u_C3R_Ctx (C function), 701
 nppiYUVToRGBBatch_8u_P3C3R_Advanced_Ctx (C function), 702
 nppiYUVToRGBBatch_8u_P3C3R_Ctx (C function), 701
 NppLibraryVersion (C struct), 27
 NppLibraryVersion.build (C var), 28
 NppLibraryVersion.major (C var), 28
 NppLibraryVersion.minor (C var), 28
 NppPointPolar (C struct), 34
 NppPointPolar.rho (C var), 34
 NppPointPolar.theta (C var), 34
 NppRoundMode (C enum), 36
 NppRoundMode.NPP_RND_FINANCIAL (C enumerator), 36
 NppRoundMode.NPP_RND_NEAR (C enumerator), 36
 NppRoundMode.NPP_RND_ZERO (C enumerator), 37
 NppRoundMode.NPP_ROUND_NEAREST_TIES_AWAY_FROM_ZERO (C enumerator), 37
 NppRoundMode.NPP_ROUND_NEAREST_TIES_TO_EVEN (C enumerator), 36
 NppRoundMode.NPP_ROUND_TOWARD_ZERO (C enumerator), 37
 npps10Log10_32s_ISfs_Ctx (C function), 1814
 npps10Log10_32s_Sfs_Ctx (C function), 1814
 nppsAbs_16s_Ctx (C function), 1794
 nppsAbs_16s_I_Ctx (C function), 1795
 nppsAbs_32f_Ctx (C function), 1795
 nppsAbs_32f_I_Ctx (C function), 1796
 nppsAbs_32s_Ctx (C function), 1795
 nppsAbs_32s_I_Ctx (C function), 1796
 nppsAbs_64f_Ctx (C function), 1795
 nppsAbs_64f_I_Ctx (C function), 1796
 nppsAdd_8u16u_Ctx (C function), 1756
 nppsAdd_8u_ISfs_Ctx (C function), 1761
 nppsAdd_8u_Sfs_Ctx (C function), 1757
 nppsAdd_16s32f_Ctx (C function), 1757
 nppsAdd_16s32s_I_Ctx (C function), 1761
 nppsAdd_16s_Ctx (C function), 1754
 nppsAdd_16s_I_Ctx (C function), 1760
 nppsAdd_16s_ISfs_Ctx (C function), 1762
 nppsAdd_16s_Sfs_Ctx (C function), 1758
 nppsAdd_16sc_ISfs_Ctx (C function), 1763

nppsAdd_16sc_Sfs_Ctx (C function), 1759
 nppsAdd_16u_Ctx (C function), 1755
 nppsAdd_16u_ISfs_Ctx (C function), 1762
 nppsAdd_16u_Sfs_Ctx (C function), 1757
 nppsAdd_32f_Ctx (C function), 1755
 nppsAdd_32f_I_Ctx (C function), 1760
 nppsAdd_32fc_Ctx (C function), 1756
 nppsAdd_32fc_I_Ctx (C function), 1760
 nppsAdd_32s_ISfs_Ctx (C function), 1762
 nppsAdd_32s_Sfs_Ctx (C function), 1758
 nppsAdd_32sc_ISfs_Ctx (C function), 1763
 nppsAdd_32sc_Sfs_Ctx (C function), 1759
 nppsAdd_32u_Ctx (C function), 1755
 nppsAdd_64f_Ctx (C function), 1755
 nppsAdd_64f_I_Ctx (C function), 1760
 nppsAdd_64fc_Ctx (C function), 1756
 nppsAdd_64fc_I_Ctx (C function), 1761
 nppsAdd_64s_Sfs_Ctx (C function), 1758
 nppsAddC_8u_ISfs_Ctx (C function), 1720
 nppsAddC_8u_Sfs_Ctx (C function), 1720
 nppsAddC_16s_ISfs_Ctx (C function), 1721
 nppsAddC_16s_Sfs_Ctx (C function), 1722
 nppsAddC_16sc_ISfs_Ctx (C function), 1722
 nppsAddC_16sc_Sfs_Ctx (C function), 1722
 nppsAddC_16u_ISfs_Ctx (C function), 1721
 nppsAddC_16u_Sfs_Ctx (C function), 1721
 nppsAddC_32f_Ctx (C function), 1724
 nppsAddC_32f_I_Ctx (C function), 1724
 nppsAddC_32fc_Ctx (C function), 1725
 nppsAddC_32fc_I_Ctx (C function), 1725
 nppsAddC_32s_ISfs_Ctx (C function), 1723
 nppsAddC_32s_Sfs_Ctx (C function), 1723
 nppsAddC_32sc_ISfs_Ctx (C function), 1723
 nppsAddC_32sc_Sfs_Ctx (C function), 1724
 nppsAddC_64f_Ctx (C function), 1726
 nppsAddC_64f_I_Ctx (C function), 1725
 nppsAddC_64fc_Ctx (C function), 1726
 nppsAddC_64fc_I_Ctx (C function), 1726
 nppsAddProduct_16s32s_Sfs_Ctx (C function), 1766
 nppsAddProduct_16s_Sfs_Ctx (C function), 1765
 nppsAddProduct_32f_Ctx (C function), 1764
 nppsAddProduct_32fc_Ctx (C function), 1764
 nppsAddProduct_32s_Sfs_Ctx (C function), 1765
 nppsAddProduct_64f_Ctx (C function), 1764
 nppsAddProduct_64fc_Ctx (C function), 1765
 nppsAddProductC_32f_Ctx (C function), 1727
 nppsAnd_8u_Ctx (C function), 1825
 nppsAnd_8u_I_Ctx (C function), 1826
 nppsAnd_16u_Ctx (C function), 1825
 nppsAnd_16u_I_Ctx (C function), 1826
 nppsAnd_32u_Ctx (C function), 1825
 nppsAnd_32u_I_Ctx (C function), 1826
 nppsAndC_8u_Ctx (C function), 1823
 nppsAndC_8u_I_Ctx (C function), 1824
 nppsAndC_16u_Ctx (C function), 1823
 nppsAndC_16u_I_Ctx (C function), 1824
 nppsAndC_32u_Ctx (C function), 1823
 nppsAndC_32u_I_Ctx (C function), 1824
 nppsArctan_32f_Ctx (C function), 1818
 nppsArctan_32f_I_Ctx (C function), 1818
 nppsArctan_64f_Ctx (C function), 1818
 nppsArctan_64f_I_Ctx (C function), 1818
 nppsAverageError_8s_Ctx (C function), 1984
 nppsAverageError_8u_Ctx (C function), 1984
 nppsAverageError_16s_Ctx (C function), 1985
 nppsAverageError_16sc_Ctx (C function), 1985
 nppsAverageError_16u_Ctx (C function), 1984
 nppsAverageError_32f_Ctx (C function), 1987
 nppsAverageError_32fc_Ctx (C function), 1988
 nppsAverageError_32s_Ctx (C function), 1986
 nppsAverageError_32sc_Ctx (C function), 1986
 nppsAverageError_32u_Ctx (C function), 1985
 nppsAverageError_64f_Ctx (C function), 1988
 nppsAverageError_64fc_Ctx (C function), 1989
 nppsAverageError_64s_Ctx (C function), 1987
 nppsAverageError_64sc_Ctx (C function), 1987
 nppsAverageErrorGetBufferSize_8s_Ctx (C function), 1989
 nppsAverageErrorGetBufferSize_8u_Ctx (C function), 1989
 nppsAverageErrorGetBufferSize_16s_Ctx (C function), 1990
 nppsAverageErrorGetBufferSize_16sc_Ctx (C function), 1990
 nppsAverageErrorGetBufferSize_16u_Ctx (C function), 1990
 nppsAverageErrorGetBufferSize_32f_Ctx (C function), 1991
 nppsAverageErrorGetBufferSize_32fc_Ctx (C function), 1992
 nppsAverageErrorGetBufferSize_32s_Ctx (C function), 1990
 nppsAverageErrorGetBufferSize_32sc_Ctx (C function), 1991
 nppsAverageErrorGetBufferSize_32u_Ctx (C function), 1990
 nppsAverageErrorGetBufferSize_64f_Ctx (C function), 1992
 nppsAverageErrorGetBufferSize_64fc_Ctx (C function), 1992

nppsAverageErrorGetBufferSize_64s_Ctx (C function), 1991	nppsAverageRelativeErrorGetBufferSize_64fc_Ctx (C function), 2012
nppsAverageErrorGetBufferSize_64sc_Ctx (C function), 1991	nppsAverageRelativeErrorGetBufferSize_64s_Ctx (C function), 2011
nppsAverageRelativeError_8s_Ctx (C function), 2003	nppsAverageRelativeErrorGetBufferSize_64sc_Ctx (C function), 2011
nppsAverageRelativeError_8u_Ctx (C function), 2003	nppsCauchy_32f_I_Ctx (C function), 1821
nppsAverageRelativeError_16s_Ctx (C function), 2004	nppsCauchyD_32f_I_Ctx (C function), 1822
nppsAverageRelativeError_16sc_Ctx (C function), 2004	nppsCauchyDD2_32f_I_Ctx (C function), 1822
nppsAverageRelativeError_16u_Ctx (C function), 2004	nppsConvert_8s16s_Ctx (C function), 1843
nppsAverageRelativeError_32f_Ctx (C function), 2007	nppsConvert_8s32f_Ctx (C function), 1843
nppsAverageRelativeError_32fc_Ctx (C function), 2007	nppsConvert_8u32f_Ctx (C function), 1844
nppsAverageRelativeError_32s_Ctx (C function), 2005	nppsConvert_16s8s_Sfs_Ctx (C function), 1844
nppsAverageRelativeError_32sc_Ctx (C function), 2006	nppsConvert_16s32f_Ctx (C function), 1844
nppsAverageRelativeError_32u_Ctx (C function), 2005	nppsConvert_16s32f_Sfs_Ctx (C function), 1847
nppsAverageRelativeError_64f_Ctx (C function), 2008	nppsConvert_16s32s_Ctx (C function), 1844
nppsAverageRelativeError_64fc_Ctx (C function), 2008	nppsConvert_16s64f_Sfs_Ctx (C function), 1847
nppsAverageRelativeError_64s_Ctx (C function), 2006	nppsConvert_16u32f_Ctx (C function), 1845
nppsAverageRelativeError_64sc_Ctx (C function), 2007	nppsConvert_32f8s_Sfs_Ctx (C function), 1848
nppsAverageRelativeErrorGetBufferSize_8s_Ctx (C function), 2009	nppsConvert_32f8u_Sfs_Ctx (C function), 1849
nppsAverageRelativeErrorGetBufferSize_8u_Ctx (C function), 2009	nppsConvert_32f16s_Sfs_Ctx (C function), 1849
nppsAverageRelativeErrorGetBufferSize_16s_Ctx (C function), 2009	nppsConvert_32f16u_Sfs_Ctx (C function), 1849
nppsAverageRelativeErrorGetBufferSize_16sc_Ctx (C function), 2010	nppsConvert_32f32s_Sfs_Ctx (C function), 1850
nppsAverageRelativeErrorGetBufferSize_16u_Ctx (C function), 2009	nppsConvert_32f64f_Ctx (C function), 1846
nppsAverageRelativeErrorGetBufferSize_32f_Ctx (C function), 2011	nppsConvert_32s16s_Ctx (C function), 1845
nppsAverageRelativeErrorGetBufferSize_32fc_Ctx (C function), 2012	nppsConvert_32s16s_Sfs_Ctx (C function), 1847
nppsAverageRelativeErrorGetBufferSize_32s_Ctx (C function), 2010	nppsConvert_32s32f_Ctx (C function), 1845
nppsAverageRelativeErrorGetBufferSize_32sc_Ctx (C function), 2011	nppsConvert_32s32f_Sfs_Ctx (C function), 1848
nppsAverageRelativeErrorGetBufferSize_32u_Ctx (C function), 2010	nppsConvert_32s64f_Ctx (C function), 1846
nppsAverageRelativeErrorGetBufferSize_64f_Ctx (C function), 2012	nppsConvert_32s64f_Sfs_Ctx (C function), 1848
	nppsConvert_64f16s_Sfs_Ctx (C function), 1850
	nppsConvert_64f32f_Ctx (C function), 1846
	nppsConvert_64f32s_Sfs_Ctx (C function), 1851
	nppsConvert_64f64s_Sfs_Ctx (C function), 1851
	nppsConvert_64s32s_Sfs_Ctx (C function), 1850
	nppsConvert_64s64f_Ctx (C function), 1846
nppsAverageRelativeErrorGetBufferSize_8u_Ctx (C function), 1880	nppsCopy_8u_Ctx (C function), 1880
nppsAverageRelativeErrorGetBufferSize_16s_Ctx (C function), 1880	nppsCopy_16s_Ctx (C function), 1880
nppsAverageRelativeErrorGetBufferSize_16sc_Ctx (C function), 1881	nppsCopy_16sc_Ctx (C function), 1881

nppsCopy_32f_Ctx (C function), 1881
 nppsCopy_32fc_Ctx (C function), 1882
 nppsCopy_32s_Ctx (C function), 1881
 nppsCopy_32sc_Ctx (C function), 1882
 nppsCopy_64fc_Ctx (C function), 1882
 nppsCopy_64s_Ctx (C function), 1881
 nppsCopy_64sc_Ctx (C function), 1882
 nppsCountInRange_32s_Ctx (C function), 1972
 nppsCountInRangeGetBufferSize_32s_Ctx (C function), 1972
 nppsCubrt_32f_Ctx (C function), 1807
 nppsCubrt_32s16s_Sfs_Ctx (C function), 1807
 nppsDiv_8u_ISfs_Ctx (C function), 1789
 nppsDiv_8u_Sfs_Ctx (C function), 1785
 nppsDiv_16s_ISfs_Ctx (C function), 1789
 nppsDiv_16s_Sfs_Ctx (C function), 1786
 nppsDiv_16sc_ISfs_Ctx (C function), 1789
 nppsDiv_16sc_Sfs_Ctx (C function), 1786
 nppsDiv_16u_ISfs_Ctx (C function), 1789
 nppsDiv_16u_Sfs_Ctx (C function), 1785
 nppsDiv_32f_Ctx (C function), 1787
 nppsDiv_32f_I_Ctx (C function), 1790
 nppsDiv_32fc_Ctx (C function), 1788
 nppsDiv_32fc_I_Ctx (C function), 1791
 nppsDiv_32s16s_Sfs_Ctx (C function), 1787
 nppsDiv_32s_ISfs_Ctx (C function), 1790
 nppsDiv_32s_Sfs_Ctx (C function), 1786
 nppsDiv_64f_Ctx (C function), 1788
 nppsDiv_64f_I_Ctx (C function), 1790
 nppsDiv_64fc_Ctx (C function), 1788
 nppsDiv_64fc_I_Ctx (C function), 1791
 nppsDiv_Round_8u_ISfs_Ctx (C function), 1793
 nppsDiv_Round_8u_Sfs_Ctx (C function), 1792
 nppsDiv_Round_16s_ISfs_Ctx (C function), 1794
 nppsDiv_Round_16s_Sfs_Ctx (C function), 1792
 nppsDiv_Round_16u_ISfs_Ctx (C function), 1793
 nppsDiv_Round_16u_Sfs_Ctx (C function), 1792
 nppsDivC_8u_ISfs_Ctx (C function), 1748
 nppsDivC_8u_Sfs_Ctx (C function), 1748
 nppsDivC_16s_ISfs_Ctx (C function), 1749
 nppsDivC_16s_Sfs_Ctx (C function), 1749
 nppsDivC_16sc_ISfs_Ctx (C function), 1750
 nppsDivC_16sc_Sfs_Ctx (C function), 1750
 nppsDivC_16u_ISfs_Ctx (C function), 1748
 nppsDivC_16u_Sfs_Ctx (C function), 1749
 nppsDivC_32f_Ctx (C function), 1751
 nppsDivC_32f_I_Ctx (C function), 1750
 nppsDivC_32fc_Ctx (C function), 1751
 nppsDivC_32fc_I_Ctx (C function), 1751
 nppsDivC_64f_Ctx (C function), 1752
 nppsDivC_64f_I_Ctx (C function), 1752
 nppsDivC_64fc_Ctx (C function), 1752
 nppsDivC_64fc_I_Ctx (C function), 1752
 nppsDivCRev_16u_Ctx (C function), 1753
 nppsDivCRev_16u_I_Ctx (C function), 1753
 nppsDivCRev_32f_Ctx (C function), 1754
 nppsDivCRev_32f_I_Ctx (C function), 1753
 nppsDotProd_16s16sc32fc_Ctx (C function), 1964
 nppsDotProd_16s16sc32sc_Sfs_Ctx (C function), 1968
 nppsDotProd_16s16sc64sc_Ctx (C function), 1962
 nppsDotProd_16s16sc_Sfs_Ctx (C function), 1969
 nppsDotProd_16s32f_Ctx (C function), 1963
 nppsDotProd_16s32s32s_Sfs_Ctx (C function), 1969
 nppsDotProd_16s32s_Sfs_Ctx (C function), 1967
 nppsDotProd_16s64s_Ctx (C function), 1961
 nppsDotProd_16s_Sfs_Ctx (C function), 1965
 nppsDotProd_16sc32fc_Ctx (C function), 1963
 nppsDotProd_16sc32sc_Sfs_Ctx (C function), 1970
 nppsDotProd_16sc64sc_Ctx (C function), 1961
 nppsDotProd_16sc_Sfs_Ctx (C function), 1965
 nppsDotProd_32f32fc64fc_Ctx (C function), 1958
 nppsDotProd_32f32fc_Ctx (C function), 1956
 nppsDotProd_32f64f_Ctx (C function), 1957
 nppsDotProd_32f_Ctx (C function), 1955
 nppsDotProd_32fc64fc_Ctx (C function), 1958
 nppsDotProd_32fc_Ctx (C function), 1956
 nppsDotProd_32s32sc_Sfs_Ctx (C function), 1971
 nppsDotProd_32s_Sfs_Ctx (C function), 1966
 nppsDotProd_32sc_Sfs_Ctx (C function), 1967
 nppsDotProd_64f64fc_Ctx (C function), 1960
 nppsDotProd_64f_Ctx (C function), 1959
 nppsDotProd_64fc_Ctx (C function), 1960
 nppsDotProdGetBufferSize_16s16sc32fc_Ctx (C function), 1964
 nppsDotProdGetBufferSize_16s16sc32sc_Sfs_Ctx (C function), 1968
 nppsDotProdGetBufferSize_16s16sc64sc_Ctx (C function), 1962
 nppsDotProdGetBufferSize_16s16sc_Sfs_Ctx (C function), 1969
 nppsDotProdGetBufferSize_16s32f_Ctx (C function), 1962
 nppsDotProdGetBufferSize_16s32s32s_Sfs_Ctx (C function), 1968

nppsDotProdGetBufferSize_16s32s_Sfs_Ctx (C function), 1967
 nppsDotProdGetBufferSize_16s64s_Ctx (C function), 1961
 nppsDotProdGetBufferSize_16s_Sfs_Ctx (C function), 1964
 nppsDotProdGetBufferSize_16sc32fc_Ctx (C function), 1963
 nppsDotProdGetBufferSize_16sc32sc_Sfs_Ctx (C function), 1970
 nppsDotProdGetBufferSize_16sc64sc_Ctx (C function), 1961
 nppsDotProdGetBufferSize_16sc_Sfs_Ctx (C function), 1965
 nppsDotProdGetBufferSize_32f32fc64fc_Ctx (C function), 1958
 nppsDotProdGetBufferSize_32f32fc_Ctx (C function), 1956
 nppsDotProdGetBufferSize_32f64f_Ctx (C function), 1957
 nppsDotProdGetBufferSize_32f_Ctx (C function), 1955
 nppsDotProdGetBufferSize_32fc64fc_Ctx (C function), 1957
 nppsDotProdGetBufferSize_32fc_Ctx (C function), 1955
 nppsDotProdGetBufferSize_32s32sc_Sfs_Ctx (C function), 1971
 nppsDotProdGetBufferSize_32s_Sfs_Ctx (C function), 1966
 nppsDotProdGetBufferSize_32sc_Sfs_Ctx (C function), 1966
 nppsDotProdGetBufferSize_64f64fc_Ctx (C function), 1960
 nppsDotProdGetBufferSize_64f_Ctx (C function), 1959
 nppsDotProdGetBufferSize_64fc_Ctx (C function), 1959
 nppsExp_16s_ISfs_Ctx (C function), 1810
 nppsExp_16s_Sfs_Ctx (C function), 1809
 nppsExp_32f64f_Ctx (C function), 1808
 nppsExp_32f_Ctx (C function), 1808
 nppsExp_32f_I_Ctx (C function), 1809
 nppsExp_32s_ISfs_Ctx (C function), 1810
 nppsExp_32s_Sfs_Ctx (C function), 1809
 nppsExp_64f_Ctx (C function), 1808
 nppsExp_64f_I_Ctx (C function), 1809
 nppsExp_64s_ISfs_Ctx (C function), 1811
 nppsExp_64s_Sfs_Ctx (C function), 1810
 nppsFree (C function), 2016
 nppsIntegral_32s_Ctx (C function), 1872
 nppsIntegralGetBufferSize_32s (C function), 1872
 nppsLn_16s_ISfs_Ctx (C function), 1813
 nppsLn_16s_Sfs_Ctx (C function), 1812
 nppsLn_32f_Ctx (C function), 1811
 nppsLn_32f_I_Ctx (C function), 1812
 nppsLn_32s16s_Sfs_Ctx (C function), 1813
 nppsLn_32s_ISfs_Ctx (C function), 1814
 nppsLn_32s_Sfs_Ctx (C function), 1813
 nppsLn_64f32f_Ctx (C function), 1812
 nppsLn_64f_Ctx (C function), 1811
 nppsLn_64f_I_Ctx (C function), 1812
 nppsLShiftC_8u_Ctx (C function), 1837
 nppsLShiftC_8u_I_Ctx (C function), 1838
 nppsLShiftC_16s_Ctx (C function), 1837
 nppsLShiftC_16s_I_Ctx (C function), 1839
 nppsLShiftC_16u_Ctx (C function), 1837
 nppsLShiftC_16u_I_Ctx (C function), 1838
 nppsLShiftC_32s_Ctx (C function), 1838
 nppsLShiftC_32s_I_Ctx (C function), 1839
 nppsLShiftC_32u_Ctx (C function), 1837
 nppsLShiftC_32u_I_Ctx (C function), 1839
 nppsMalloc_8s (C function), 2013
 nppsMalloc_8u (C function), 2013
 nppsMalloc_16s (C function), 2014
 nppsMalloc_16sc (C function), 2014
 nppsMalloc_16u (C function), 2013
 nppsMalloc_32f (C function), 2014
 nppsMalloc_32fc (C function), 2015
 nppsMalloc_32s (C function), 2014
 nppsMalloc_32sc (C function), 2014
 nppsMalloc_32u (C function), 2014
 nppsMalloc_64f (C function), 2015
 nppsMalloc_64fc (C function), 2015
 nppsMalloc_64s (C function), 2015
 nppsMalloc_64sc (C function), 2015
 nppsMax_16s_Ctx (C function), 1894
 nppsMax_32f_Ctx (C function), 1894
 nppsMax_32s_Ctx (C function), 1894
 nppsMax_64f_Ctx (C function), 1895
 nppsMaxAbs_16s_Ctx (C function), 1898
 nppsMaxAbs_32s_Ctx (C function), 1898
 nppsMaxAbsGetBufferSize_16s_Ctx (C function), 1898
 nppsMaxAbsGetBufferSize_32s_Ctx (C function), 1898
 nppsMaxAbsIndx_16s_Ctx (C function), 1899
 nppsMaxAbsIndx_32s_Ctx (C function), 1900
 nppsMaxAbsIndxGetBufferSize_16s_Ctx (C function), 1899
 nppsMaxAbsIndxGetBufferSize_32s_Ctx (C function), 1899
 nppsMaxEvery_8u_I_Ctx (C function), 1885
 nppsMaxEvery_16s_I_Ctx (C function), 1885
 nppsMaxEvery_16u_I_Ctx (C function), 1885
 nppsMaxEvery_32f_I_Ctx (C function), 1886
 nppsMaxEvery_32s_I_Ctx (C function), 1886

nppsMaxGetBufferSize_16s_Ctx (C function), 1893
 nppsMaxGetBufferSize_32f_Ctx (C function), 1893
 nppsMaxGetBufferSize_32s_Ctx (C function), 1893
 nppsMaxGetBufferSize_64f_Ctx (C function), 1893
 nppsMaximumError_8s_Ctx (C function), 1974
 nppsMaximumError_8u_Ctx (C function), 1974
 nppsMaximumError_16s_Ctx (C function), 1975
 nppsMaximumError_16sc_Ctx (C function), 1976
 nppsMaximumError_16u_Ctx (C function), 1975
 nppsMaximumError_32f_Ctx (C function), 1978
 nppsMaximumError_32fc_Ctx (C function), 1978
 nppsMaximumError_32s_Ctx (C function), 1976
 nppsMaximumError_32sc_Ctx (C function), 1977
 nppsMaximumError_32u_Ctx (C function), 1976
 nppsMaximumError_64f_Ctx (C function), 1979
 nppsMaximumError_64fc_Ctx (C function), 1979
 nppsMaximumError_64s_Ctx (C function), 1977
 nppsMaximumError_64sc_Ctx (C function), 1978
 nppsMaximumErrorGetBufferSize_8s_Ctx (C function), 1980
 nppsMaximumErrorGetBufferSize_8u_Ctx (C function), 1980
 nppsMaximumErrorGetBufferSize_16s_Ctx (C function), 1980
 nppsMaximumErrorGetBufferSize_16sc_Ctx (C function), 1981
 nppsMaximumErrorGetBufferSize_16u_Ctx (C function), 1980
 nppsMaximumErrorGetBufferSize_32f_Ctx (C function), 1982
 nppsMaximumErrorGetBufferSize_32fc_Ctx (C function), 1982
 nppsMaximumErrorGetBufferSize_32s_Ctx (C function), 1981
 nppsMaximumErrorGetBufferSize_32sc_Ctx (C function), 1981
 nppsMaximumErrorGetBufferSize_32u_Ctx (C function), 1981
 nppsMaximumErrorGetBufferSize_64f_Ctx (C function), 1983
 nppsMaximumErrorGetBufferSize_64fc_Ctx (C function), 1983
 nppsMaximumErrorGetBufferSize_64s_Ctx (C function), 1982
 nppsMaximumErrorGetBufferSize_64sc_Ctx (C function), 1982
 nppsMaximumRelativeError_8s_Ctx (C function), 1993
 nppsMaximumRelativeError_8u_Ctx (C function), 1993
 nppsMaximumRelativeError_16s_Ctx (C function), 1994
 nppsMaximumRelativeError_16sc_Ctx (C function), 1994
 nppsMaximumRelativeError_16u_Ctx (C function), 1994
 nppsMaximumRelativeError_32f_Ctx (C function), 1997
 nppsMaximumRelativeError_32fc_Ctx (C function), 1997
 nppsMaximumRelativeError_32s_Ctx (C function), 1995
 nppsMaximumRelativeError_32sc_Ctx (C function), 1996
 nppsMaximumRelativeError_32u_Ctx (C function), 1995
 nppsMaximumRelativeError_64f_Ctx (C function), 1998
 nppsMaximumRelativeError_64fc_Ctx (C function), 1998
 nppsMaximumRelativeError_64s_Ctx (C function), 1996
 nppsMaximumRelativeError_64sc_Ctx (C function), 1996
 nppsMaximumRelativeErrorGetBufferSize_8s_Ctx (C function), 1999
 nppsMaximumRelativeErrorGetBufferSize_8u_Ctx (C function), 1999
 nppsMaximumRelativeErrorGetBufferSize_16s_Ctx (C function), 1999
 nppsMaximumRelativeErrorGetBufferSize_16sc_Ctx (C function), 2000
 nppsMaximumRelativeErrorGetBufferSize_16u_Ctx (C function), 1999
 nppsMaximumRelativeErrorGetBufferSize_32f_Ctx (C function), 2001
 nppsMaximumRelativeErrorGetBufferSize_32fc_Ctx (C function), 2002
 nppsMaximumRelativeErrorGetBufferSize_32s_Ctx (C function), 2000
 nppsMaximumRelativeErrorGetBufferSize_32sc_Ctx (C function), 2001
 nppsMaximumRelativeErrorGetBufferSize_32u_Ctx (C function), 2000
 nppsMaximumRelativeErrorGetBufferSize_64f_Ctx (C function), 2002
 nppsMaximumRelativeErrorGetBufferSize_64fc_Ctx (C function), 2002
 nppsMaximumRelativeErrorGetBufferSize_64s_Ctx

(C function), 2001
 nppsMaximumRelativeErrorGetBufferSize_64f_Ctx (C function), 2001
 (C function), 2001
 nppsMaxIndx_16s_Ctx (C function), 1896
 nppsMaxIndx_32f_Ctx (C function), 1897
 nppsMaxIndx_32s_Ctx (C function), 1896
 nppsMaxIndx_64f_Ctx (C function), 1897
 nppsMaxIndxGetBufferSize_16s_Ctx (C function), 1895
 nppsMaxIndxGetBufferSize_32f_Ctx (C function), 1895
 nppsMaxIndxGetBufferSize_32s_Ctx (C function), 1895
 nppsMaxIndxGetBufferSize_64f_Ctx (C function), 1896
 nppsMean_16s_Sfs_Ctx (C function), 1911
 nppsMean_16sc_Sfs_Ctx (C function), 1912
 nppsMean_32f_Ctx (C function), 1910
 nppsMean_32fc_Ctx (C function), 1910
 nppsMean_32s_Sfs_Ctx (C function), 1912
 nppsMean_64f_Ctx (C function), 1911
 nppsMean_64fc_Ctx (C function), 1911
 nppsMeanGetBufferSize_16s_Sfs_Ctx (C function), 1909
 nppsMeanGetBufferSize_16sc_Sfs_Ctx (C function), 1910
 nppsMeanGetBufferSize_32f_Ctx (C function), 1908
 nppsMeanGetBufferSize_32fc_Ctx (C function), 1908
 nppsMeanGetBufferSize_32s_Sfs_Ctx (C function), 1909
 nppsMeanGetBufferSize_64f_Ctx (C function), 1909
 nppsMeanGetBufferSize_64fc_Ctx (C function), 1909
 nppsMeanStdDev_16s32s_Sfs_Ctx (C function), 1917
 nppsMeanStdDev_16s_Sfs_Ctx (C function), 1918
 nppsMeanStdDev_32f_Ctx (C function), 1917
 nppsMeanStdDev_64f_Ctx (C function), 1917
 nppsMeanStdDevGetBufferSize_16s32s_Sfs_Ctx (C function), 1916
 nppsMeanStdDevGetBufferSize_16s_Sfs_Ctx (C function), 1916
 nppsMeanStdDevGetBufferSize_32f_Ctx (C function), 1916
 nppsMeanStdDevGetBufferSize_64f_Ctx (C function), 1916
 nppsMin_16s_Ctx (C function), 1901
 nppsMin_32f_Ctx (C function), 1902
 nppsMin_32s_Ctx (C function), 1902
 nppsMin_64f_Ctx (C function), 1902
 nppsMinAbs_16s_Ctx (C function), 1906
 nppsMinAbs_32s_Ctx (C function), 1906
 nppsMinAbsGetBufferSize_16s_Ctx (C function), 1905
 nppsMinAbsGetBufferSize_32s_Ctx (C function), 1906
 nppsMinAbsIndx_16s_Ctx (C function), 1907
 nppsMinAbsIndx_32s_Ctx (C function), 1907
 nppsMinAbsIndxGetBufferSize_16s_Ctx (C function), 1907
 nppsMinAbsIndxGetBufferSize_32s_Ctx (C function), 1907
 nppsMinEvery_8u_I_Ctx (C function), 1883
 nppsMinEvery_16s_I_Ctx (C function), 1884
 nppsMinEvery_16u_I_Ctx (C function), 1883
 nppsMinEvery_32f_I_Ctx (C function), 1884
 nppsMinEvery_32s_I_Ctx (C function), 1884
 nppsMinEvery_64f_I_Ctx (C function), 1885
 nppsMinGetBufferSize_16s_Ctx (C function), 1900
 nppsMinGetBufferSize_32f_Ctx (C function), 1901
 nppsMinGetBufferSize_32s_Ctx (C function), 1901
 nppsMinGetBufferSize_64f_Ctx (C function), 1901
 nppsMinIndx_16s_Ctx (C function), 1904
 nppsMinIndx_32f_Ctx (C function), 1905
 nppsMinIndx_32s_Ctx (C function), 1904
 nppsMinIndx_64f_Ctx (C function), 1905
 nppsMinIndxGetBufferSize_16s_Ctx (C function), 1903
 nppsMinIndxGetBufferSize_32f_Ctx (C function), 1903
 nppsMinIndxGetBufferSize_32s_Ctx (C function), 1903
 nppsMinIndxGetBufferSize_64f_Ctx (C function), 1904
 nppsMinMax_8u_Ctx (C function), 1920
 nppsMinMax_16s_Ctx (C function), 1921
 nppsMinMax_16u_Ctx (C function), 1921
 nppsMinMax_32f_Ctx (C function), 1922
 nppsMinMax_32s_Ctx (C function), 1922
 nppsMinMax_32u_Ctx (C function), 1921
 nppsMinMax_64f_Ctx (C function), 1923
 nppsMinMaxGetBufferSize_8u_Ctx (C function), 1919
 nppsMinMaxGetBufferSize_16s_Ctx (C function), 1919
 nppsMinMaxGetBufferSize_16u_Ctx (C function), 1919
 nppsMinMaxGetBufferSize_32f_Ctx (C function), 1920

nppsMinMaxGetBufferSize_32s_Ctx (C function), 1919
 nppsMinMaxGetBufferSize_32u_Ctx (C function), 1920
 nppsMinMaxGetBufferSize_64f_Ctx (C function), 1920
 nppsMinMaxIndx_8u_Ctx (C function), 1925
 nppsMinMaxIndx_16s_Ctx (C function), 1925
 nppsMinMaxIndx_16u_Ctx (C function), 1926
 nppsMinMaxIndx_32f_Ctx (C function), 1927
 nppsMinMaxIndx_32s_Ctx (C function), 1926
 nppsMinMaxIndx_32u_Ctx (C function), 1927
 nppsMinMaxIndx_64f_Ctx (C function), 1927
 nppsMinMaxIndxGetBufferSize_8u_Ctx (C function), 1923
 nppsMinMaxIndxGetBufferSize_16s_Ctx (C function), 1923
 nppsMinMaxIndxGetBufferSize_16u_Ctx (C function), 1923
 nppsMinMaxIndxGetBufferSize_32f_Ctx (C function), 1924
 nppsMinMaxIndxGetBufferSize_32s_Ctx (C function), 1924
 nppsMinMaxIndxGetBufferSize_32u_Ctx (C function), 1924
 nppsMinMaxIndxGetBufferSize_64f_Ctx (C function), 1924
 nppsMul_8u16u_Ctx (C function), 1768
 nppsMul_8u_ISfs_Ctx (C function), 1775
 nppsMul_8u_Sfs_Ctx (C function), 1769
 nppsMul_16s32f_Ctx (C function), 1768
 nppsMul_16s32s_Sfs_Ctx (C function), 1772
 nppsMul_16s_Ctx (C function), 1767
 nppsMul_16s_I_Ctx (C function), 1773
 nppsMul_16s_ISfs_Ctx (C function), 1775
 nppsMul_16s_Sfs_Ctx (C function), 1770
 nppsMul_16sc_ISfs_Ctx (C function), 1776
 nppsMul_16sc_Sfs_Ctx (C function), 1771
 nppsMul_16u16s_Sfs_Ctx (C function), 1771
 nppsMul_16u_ISfs_Ctx (C function), 1775
 nppsMul_16u_Sfs_Ctx (C function), 1770
 nppsMul_32f32fc_Ctx (C function), 1769
 nppsMul_32f32fc_I_Ctx (C function), 1774
 nppsMul_32f_Ctx (C function), 1767
 nppsMul_32f_I_Ctx (C function), 1773
 nppsMul_32fc_Ctx (C function), 1767
 nppsMul_32fc_I_Ctx (C function), 1774
 nppsMul_32s32sc_ISfs_Ctx (C function), 1777
 nppsMul_32s32sc_Sfs_Ctx (C function), 1772
 nppsMul_32s_ISfs_Ctx (C function), 1776
 nppsMul_32s_Sfs_Ctx (C function), 1770
 nppsMul_32sc_ISfs_Ctx (C function), 1776
 nppsMul_32sc_Sfs_Ctx (C function), 1771
 nppsMul_64f_Ctx (C function), 1767
 nppsMul_64f_I_Ctx (C function), 1773
 nppsMul_64fc_Ctx (C function), 1768
 nppsMul_64fc_I_Ctx (C function), 1774
 nppsMul_Low_32s_Sfs_Ctx (C function), 1773
 nppsMulC_8u_ISfs_Ctx (C function), 1727
 nppsMulC_8u_Sfs_Ctx (C function), 1728
 nppsMulC_16s_ISfs_Ctx (C function), 1729
 nppsMulC_16s_Sfs_Ctx (C function), 1729
 nppsMulC_16sc_ISfs_Ctx (C function), 1729
 nppsMulC_16sc_Sfs_Ctx (C function), 1729
 nppsMulC_16u_ISfs_Ctx (C function), 1728
 nppsMulC_16u_Sfs_Ctx (C function), 1728
 nppsMulC_32f16s_Sfs_Ctx (C function), 1732
 nppsMulC_32f_Ctx (C function), 1731
 nppsMulC_32f_I_Ctx (C function), 1731
 nppsMulC_32fc_Ctx (C function), 1733
 nppsMulC_32fc_I_Ctx (C function), 1732
 nppsMulC_32s_ISfs_Ctx (C function), 1730
 nppsMulC_32s_Sfs_Ctx (C function), 1730
 nppsMulC_32sc_ISfs_Ctx (C function), 1730
 nppsMulC_32sc_Sfs_Ctx (C function), 1731
 nppsMulC_64f64s_ISfs_Ctx (C function), 1734
 nppsMulC_64f_Ctx (C function), 1733
 nppsMulC_64f_I_Ctx (C function), 1733
 nppsMulC_64fc_Ctx (C function), 1734
 nppsMulC_64fc_I_Ctx (C function), 1734
 nppsMulC_Low_32f16s_Ctx (C function), 1732
 nppsNorm_Inf_16s32f_Ctx (C function), 1930
 nppsNorm_Inf_16s32s_Sfs_Ctx (C function), 1931
 nppsNorm_Inf_32f_Ctx (C function), 1928
 nppsNorm_Inf_32fc32f_Ctx (C function), 1930
 nppsNorm_Inf_64f_Ctx (C function), 1929
 nppsNorm_Inf_64fc64f_Ctx (C function), 1931
 nppsNorm_L1_16s32f_Ctx (C function), 1933
 nppsNorm_L1_16s32s_Sfs_Ctx (C function), 1935
 nppsNorm_L1_16s64s_Sfs_Ctx (C function), 1936
 nppsNorm_L1_32f_Ctx (C function), 1932
 nppsNorm_L1_32fc64f_Ctx (C function), 1934
 nppsNorm_L1_64f_Ctx (C function), 1933
 nppsNorm_L1_64fc64f_Ctx (C function), 1935
 nppsNorm_L2_16s32f_Ctx (C function), 1938
 nppsNorm_L2_16s32s_Sfs_Ctx (C function), 1940
 nppsNorm_L2_32f_Ctx (C function), 1937
 nppsNorm_L2_32fc64f_Ctx (C function), 1939
 nppsNorm_L2_64f_Ctx (C function), 1937
 nppsNorm_L2_64fc64f_Ctx (C function), 1939
 nppsNorm_L2Sqr_16s64s_Sfs_Ctx (C function), 1940
 nppsNormalize_16s_Sfs_Ctx (C function), 1820

nppsNormalize_16sc_Sfs_Ctx (C function), 1821
 nppsNormalize_32f_Ctx (C function), 1819
 nppsNormalize_32fc_Ctx (C function), 1819
 nppsNormalize_64f_Ctx (C function), 1820
 nppsNormalize_64fc_Ctx (C function), 1820
 nppsNormDiff_Inf_16s32f_Ctx (C function), 1943
 nppsNormDiff_Inf_16s32s_Sfs_Ctx (C function), 1944
 nppsNormDiff_Inf_32f_Ctx (C function), 1941
 nppsNormDiff_Inf_32fc32f_Ctx (C function), 1943
 nppsNormDiff_Inf_64f_Ctx (C function), 1942
 nppsNormDiff_Inf_64fc64f_Ctx (C function), 1944
 nppsNormDiff_L1_16s32f_Ctx (C function), 1947
 nppsNormDiff_L1_16s32s_Sfs_Ctx (C function), 1949
 nppsNormDiff_L1_16s64s_Sfs_Ctx (C function), 1949
 nppsNormDiff_L1_32f_Ctx (C function), 1945
 nppsNormDiff_L1_32fc64f_Ctx (C function), 1947
 nppsNormDiff_L1_64f_Ctx (C function), 1946
 nppsNormDiff_L1_64fc64f_Ctx (C function), 1948
 nppsNormDiff_L2_16s32f_Ctx (C function), 1951
 nppsNormDiff_L2_16s32s_Sfs_Ctx (C function), 1953
 nppsNormDiff_L2_32f_Ctx (C function), 1950
 nppsNormDiff_L2_32fc64f_Ctx (C function), 1952
 nppsNormDiff_L2_64f_Ctx (C function), 1951
 nppsNormDiff_L2_64fc64f_Ctx (C function), 1953
 nppsNormDiff_L2Sqr_16s64s_Sfs_Ctx (C function), 1954
 nppsNormDiffInfGetBufferSize_16s32f_Ctx (C function), 1942
 nppsNormDiffInfGetBufferSize_16s32s_Sfs_Ctx (C function), 1944
 nppsNormDiffInfGetBufferSize_32f_Ctx (C function), 1941
 nppsNormDiffInfGetBufferSize_32fc32f_Ctx (C function), 1943
 nppsNormDiffInfGetBufferSize_64f_Ctx (C function), 1942
 nppsNormDiffInfGetBufferSize_64fc64f_Ctx (C function), 1944
 nppsNormDiffL1GetBufferSize_16s32f_Ctx (C function), 1946
 nppsNormDiffL1GetBufferSize_16s32s_Sfs_Ctx (C function), 1948
 nppsNormDiffL1GetBufferSize_16s64s_Sfs_Ctx (C function), 1949
 nppsNormDiffL1GetBufferSize_32f_Ctx (C function), 1945
 nppsNormDiffL1GetBufferSize_32fc64f_Ctx (C function), 1947
 nppsNormDiffL1GetBufferSize_64f_Ctx (C function), 1946
 nppsNormDiffL1GetBufferSize_64fc64f_Ctx (C function), 1948
 nppsNormDiffL2GetBufferSize_16s32f_Ctx (C function), 1951
 nppsNormDiffL2GetBufferSize_16s32s_Sfs_Ctx (C function), 1953
 nppsNormDiffL2GetBufferSize_32f_Ctx (C function), 1950
 nppsNormDiffL2GetBufferSize_32fc64f_Ctx (C function), 1952
 nppsNormDiffL2GetBufferSize_64f_Ctx (C function), 1951
 nppsNormDiffL2GetBufferSize_64fc64f_Ctx (C function), 1952
 nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs_Ctx (C function), 1954
 nppsNormInfGetBufferSize_16s32f_Ctx (C function), 1929
 nppsNormInfGetBufferSize_16s32s_Sfs_Ctx (C function), 1931
 nppsNormInfGetBufferSize_32f_Ctx (C function), 1928
 nppsNormInfGetBufferSize_32fc32f_Ctx (C function), 1930
 nppsNormInfGetBufferSize_64f_Ctx (C function), 1929
 nppsNormInfGetBufferSize_64fc64f_Ctx (C function), 1931
 nppsNormL1GetBufferSize_16s32f_Ctx (C function), 1933
 nppsNormL1GetBufferSize_16s32s_Sfs_Ctx (C function), 1935
 nppsNormL1GetBufferSize_16s64s_Sfs_Ctx (C function), 1936
 nppsNormL1GetBufferSize_32f_Ctx (C function), 1932
 nppsNormL1GetBufferSize_32fc64f_Ctx (C function), 1934
 nppsNormL1GetBufferSize_64f_Ctx (C function), 1933
 nppsNormL1GetBufferSize_64fc64f_Ctx (C function), 1934
 nppsNormL2GetBufferSize_16s32f_Ctx (C function), 1938

nppsNormL2GetBufferSize_16s32s_Sfs_Ctx (C function), 1939
 nppsNormL2GetBufferSize_32f_Ctx (C function), 1937
 nppsNormL2GetBufferSize_32fc64f_Ctx (C function), 1938
 nppsNormL2GetBufferSize_64f_Ctx (C function), 1937
 nppsNormL2GetBufferSize_64fc64f_Ctx (C function), 1939
 nppsNormL2SqrGetBufferSize_16s64s_Sfs_Ctx (C function), 1940
 nppsNot_8u_Ctx (C function), 1835
 nppsNot_8u_I_Ctx (C function), 1835
 nppsNot_16u_Ctx (C function), 1835
 nppsNot_16u_I_Ctx (C function), 1836
 nppsNot_32u_Ctx (C function), 1835
 nppsNot_32u_I_Ctx (C function), 1836
 nppsOr_8u_Ctx (C function), 1829
 nppsOr_8u_I_Ctx (C function), 1830
 nppsOr_16u_Ctx (C function), 1829
 nppsOr_16u_I_Ctx (C function), 1830
 nppsOr_32u_Ctx (C function), 1829
 nppsOr_32u_I_Ctx (C function), 1830
 nppsOrC_8u_Ctx (C function), 1827
 nppsOrC_8u_I_Ctx (C function), 1828
 nppsOrC_16u_Ctx (C function), 1827
 nppsOrC_16u_I_Ctx (C function), 1828
 nppsOrC_32u_Ctx (C function), 1827
 nppsOrC_32u_I_Ctx (C function), 1828
 nppsRShiftC_8u_Ctx (C function), 1840
 nppsRShiftC_8u_I_Ctx (C function), 1841
 nppsRShiftC_16s_Ctx (C function), 1840
 nppsRShiftC_16s_I_Ctx (C function), 1842
 nppsRShiftC_16u_Ctx (C function), 1840
 nppsRShiftC_16u_I_Ctx (C function), 1841
 nppsRShiftC_32s_Ctx (C function), 1841
 nppsRShiftC_32s_I_Ctx (C function), 1842
 nppsRShiftC_32u_Ctx (C function), 1841
 nppsRShiftC_32u_I_Ctx (C function), 1842
 nppsSet_8s_Ctx (C function), 1873
 nppsSet_8u_Ctx (C function), 1873
 nppsSet_16s_Ctx (C function), 1874
 nppsSet_16sc_Ctx (C function), 1874
 nppsSet_16u_Ctx (C function), 1874
 nppsSet_32f_Ctx (C function), 1875
 nppsSet_32fc_Ctx (C function), 1875
 nppsSet_32s_Ctx (C function), 1875
 nppsSet_32sc_Ctx (C function), 1875
 nppsSet_32u_Ctx (C function), 1874
 nppsSet_64f_Ctx (C function), 1876
 nppsSet_64fc_Ctx (C function), 1877
 nppsSet_64s_Ctx (C function), 1876
 nppsSet_64sc_Ctx (C function), 1876
 nppsSqr_8u_ISfs_Ctx (C function), 1800
 nppsSqr_8u_Sfs_Ctx (C function), 1799
 nppsSqr_16s_ISfs_Ctx (C function), 1800
 nppsSqr_16s_Sfs_Ctx (C function), 1799
 nppsSqr_16sc_ISfs_Ctx (C function), 1801
 nppsSqr_16sc_Sfs_Ctx (C function), 1800
 nppsSqr_16u_ISfs_Ctx (C function), 1800
 nppsSqr_16u_Sfs_Ctx (C function), 1799
 nppsSqr_32f_Ctx (C function), 1797
 nppsSqr_32f_I_Ctx (C function), 1798
 nppsSqr_32fc_Ctx (C function), 1797
 nppsSqr_32fc_I_Ctx (C function), 1798
 nppsSqr_64f_Ctx (C function), 1797
 nppsSqr_64f_I_Ctx (C function), 1798
 nppsSqr_64fc_Ctx (C function), 1797
 nppsSqr_64fc_I_Ctx (C function), 1799
 nppsSqrt_8u_ISfs_Ctx (C function), 1805
 nppsSqrt_8u_Sfs_Ctx (C function), 1803
 nppsSqrt_16s_ISfs_Ctx (C function), 1806
 nppsSqrt_16s_Sfs_Ctx (C function), 1804
 nppsSqrt_16sc_ISfs_Ctx (C function), 1806
 nppsSqrt_16sc_Sfs_Ctx (C function), 1804
 nppsSqrt_16u_ISfs_Ctx (C function), 1806
 nppsSqrt_16u_Sfs_Ctx (C function), 1804
 nppsSqrt_32f_Ctx (C function), 1801
 nppsSqrt_32f_I_Ctx (C function), 1802
 nppsSqrt_32fc_Ctx (C function), 1802
 nppsSqrt_32fc_I_Ctx (C function), 1803
 nppsSqrt_32s16s_Sfs_Ctx (C function), 1805
 nppsSqrt_64f_Ctx (C function), 1802
 nppsSqrt_64f_I_Ctx (C function), 1803
 nppsSqrt_64fc_Ctx (C function), 1802
 nppsSqrt_64fc_I_Ctx (C function), 1803
 nppsSqrt_64s16s_Sfs_Ctx (C function), 1805
 nppsSqrt_64s_ISfs_Ctx (C function), 1807
 nppsSqrt_64s_Sfs_Ctx (C function), 1805
 nppsStdDev_16s32s_Sfs_Ctx (C function), 1914
 nppsStdDev_16s_Sfs_Ctx (C function), 1915
 nppsStdDev_32f_Ctx (C function), 1914
 nppsStdDev_64f_Ctx (C function), 1914
 nppsStdDevGetBufferSize_16s32s_Sfs_Ctx (C function), 1913
 nppsStdDevGetBufferSize_16s_Sfs_Ctx (C function), 1913
 nppsStdDevGetBufferSize_32f_Ctx (C function), 1913
 nppsStdDevGetBufferSize_64f_Ctx (C function), 1913
 nppsSub_8u_ISfs_Ctx (C function), 1783
 nppsSub_8u_Sfs_Ctx (C function), 1779
 nppsSub_16s32f_Ctx (C function), 1779
 nppsSub_16s_Ctx (C function), 1777
 nppsSub_16s_I_Ctx (C function), 1781

nppsSub_16s_ISfs_Ctx (C function), 1783
 nppsSub_16s_Sfs_Ctx (C function), 1780
 nppsSub_16sc_ISfs_Ctx (C function), 1784
 nppsSub_16sc_Sfs_Ctx (C function), 1781
 nppsSub_16u_ISfs_Ctx (C function), 1783
 nppsSub_16u_Sfs_Ctx (C function), 1779
 nppsSub_32f_Ctx (C function), 1777
 nppsSub_32f_I_Ctx (C function), 1782
 nppsSub_32fc_Ctx (C function), 1778
 nppsSub_32fc_I_Ctx (C function), 1782
 nppsSub_32s_ISfs_Ctx (C function), 1784
 nppsSub_32s_Sfs_Ctx (C function), 1780
 nppsSub_32sc_ISfs_Ctx (C function), 1784
 nppsSub_32sc_Sfs_Ctx (C function), 1781
 nppsSub_64f_Ctx (C function), 1778
 nppsSub_64f_I_Ctx (C function), 1782
 nppsSub_64fc_Ctx (C function), 1778
 nppsSub_64fc_I_Ctx (C function), 1783
 nppsSubC_8u_ISfs_Ctx (C function), 1735
 nppsSubC_8u_Sfs_Ctx (C function), 1735
 nppsSubC_16s_ISfs_Ctx (C function), 1736
 nppsSubC_16s_Sfs_Ctx (C function), 1736
 nppsSubC_16sc_ISfs_Ctx (C function), 1737
 nppsSubC_16sc_Sfs_Ctx (C function), 1737
 nppsSubC_16u_ISfs_Ctx (C function), 1735
 nppsSubC_16u_Sfs_Ctx (C function), 1736
 nppsSubC_32f_Ctx (C function), 1739
 nppsSubC_32f_I_Ctx (C function), 1739
 nppsSubC_32fc_Ctx (C function), 1740
 nppsSubC_32fc_I_Ctx (C function), 1739
 nppsSubC_32s_ISfs_Ctx (C function), 1737
 nppsSubC_32s_Sfs_Ctx (C function), 1738
 nppsSubC_32sc_ISfs_Ctx (C function), 1738
 nppsSubC_32sc_Sfs_Ctx (C function), 1738
 nppsSubC_64f_Ctx (C function), 1740
 nppsSubC_64f_I_Ctx (C function), 1740
 nppsSubC_64fc_Ctx (C function), 1741
 nppsSubC_64fc_I_Ctx (C function), 1740
 nppsSubCRev_8u_ISfs_Ctx (C function), 1741
 nppsSubCRev_8u_Sfs_Ctx (C function), 1741
 nppsSubCRev_16s_ISfs_Ctx (C function), 1742
 nppsSubCRev_16s_Sfs_Ctx (C function), 1743
 nppsSubCRev_16sc_ISfs_Ctx (C function), 1743
 nppsSubCRev_16sc_Sfs_Ctx (C function), 1743
 nppsSubCRev_16u_ISfs_Ctx (C function), 1742
 nppsSubCRev_16u_Sfs_Ctx (C function), 1742
 nppsSubCRev_32f_Ctx (C function), 1745
 nppsSubCRev_32f_I_Ctx (C function), 1745
 nppsSubCRev_32fc_Ctx (C function), 1746
 nppsSubCRev_32fc_I_Ctx (C function), 1746
 nppsSubCRev_32s_ISfs_Ctx (C function), 1744
 nppsSubCRev_32s_Sfs_Ctx (C function), 1744
 nppsSubCRev_32sc_ISfs_Ctx (C function), 1745
 nppsSubCRev_32sc_Sfs_Ctx (C function), 1745
 nppsSubCRev_64f_Ctx (C function), 1747
 nppsSubCRev_64f_I_Ctx (C function), 1746
 nppsSubCRev_64fc_Ctx (C function), 1747
 nppsSubCRev_64fc_I_Ctx (C function), 1747
 nppsSum_16s32s_Sfs_Ctx (C function), 1892
 nppsSum_16s_Sfs_Ctx (C function), 1890
 nppsSum_16sc32sc_Sfs_Ctx (C function), 1891
 nppsSum_16sc_Sfs_Ctx (C function), 1891
 nppsSum_32f_Ctx (C function), 1889
 nppsSum_32fc_Ctx (C function), 1889
 nppsSum_32s_Sfs_Ctx (C function), 1891
 nppsSum_64f_Ctx (C function), 1890
 nppsSum_64fc_Ctx (C function), 1890
 nppsSumGetBufferSize_16s32s_Sfs_Ctx (C function), 1889
 nppsSumGetBufferSize_16s_Sfs_Ctx (C function), 1888
 nppsSumGetBufferSize_16sc32sc_Sfs_Ctx (C function), 1888
 nppsSumGetBufferSize_16sc_Sfs_Ctx (C function), 1888
 nppsSumGetBufferSize_32f_Ctx (C function), 1887
 nppsSumGetBufferSize_32fc_Ctx (C function), 1887
 nppsSumGetBufferSize_32s_Sfs_Ctx (C function), 1888
 nppsSumGetBufferSize_64f_Ctx (C function), 1887
 nppsSumGetBufferSize_64fc_Ctx (C function), 1887
 nppsSumLn_16s32f_Ctx (C function), 1817
 nppsSumLn_32f64f_Ctx (C function), 1816
 nppsSumLn_32f_Ctx (C function), 1815
 nppsSumLn_64f_Ctx (C function), 1816
 nppsSumLnGetBufferSize_16s32f_Ctx (C function), 1817
 nppsSumLnGetBufferSize_32f64f_Ctx (C function), 1816
 nppsSumLnGetBufferSize_32f_Ctx (C function), 1815
 nppsSumLnGetBufferSize_64f_Ctx (C function), 1816
 NppStatus (C enum), 23
 NppStatus.NPP_AFFINE_QUAD_INCORRECT_WARNING (C enumerator), 27
 NppStatus.NPP_ALIGNMENT_ERROR (C enumerator), 24
 NppStatus.NPP_ANCHOR_ERROR (C enumerator), 25

NppStatus.NPP_BAD_ARGUMENT_ERROR (C enumerator), 26
 NppStatus.NPP_CHANNEL_ERROR (C enumerator), 25
 NppStatus.NPP_CHANNEL_ORDER_ERROR (C enumerator), 25
 NppStatus.NPP_COEFFICIENT_ERROR (C enumerator), 25
 NppStatus.NPP_COI_ERROR (C enumerator), 25
 NppStatus.NPP_CONTEXT_MATCH_ERROR (C enumerator), 26
 NppStatus.NPP_CORRUPTED_DATA_ERROR (C enumerator), 25
 NppStatus.NPP_CUDA_KERNEL_EXECUTION_ERROR (C enumerator), 24
 NppStatus.NPP_DATA_TYPE_ERROR (C enumerator), 25
 NppStatus.NPP_DIVIDE_BY_ZERO_ERROR (C enumerator), 26
 NppStatus.NPP_DIVIDE_BY_ZERO_WARNING (C enumerator), 27
 NppStatus.NPP_DIVISOR_ERROR (C enumerator), 25
 NppStatus.NPP_DOUBLE_SIZE_WARNING (C enumerator), 27
 NppStatus.NPP_ERROR (C enumerator), 27
 NppStatus.NPP_ERROR_RESERVED (C enumerator), 27
 NppStatus.NPP_FFT_FLAG_ERROR (C enumerator), 26
 NppStatus.NPP_FFT_ORDER_ERROR (C enumerator), 26
 NppStatus.NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR (C enumerator), 24
 NppStatus.NPP_HISTOGRAM_NUMBER_OF_LEVELS_ERROR (C enumerator), 24
 NppStatus.NPP_INTERPOLATION_ERROR (C enumerator), 26
 NppStatus.NPP_INVALID_DEVICE_POINTER_ERROR (C enumerator), 23
 NppStatus.NPP_INVALID_HOST_POINTER_ERROR (C enumerator), 23
 NppStatus.NPP_LIBRARY_VERSION_MISMATCH_ERROR (C enumerator), 23
 NppStatus.NPP_LUT_NUMBER_OF_LEVELS_ERROR (C enumerator), 24
 NppStatus.NPP_LUT_PALETTE_BITSIZE_ERROR (C enumerator), 23
 NppStatus.NPP_MASK_SIZE_ERROR (C enumerator), 25
 NppStatus.NPP_MEMCPY_ERROR (C enumerator), 24
 NppStatus.NPP_MEMFREE_ERROR (C enumerator), 24
 NppStatus.NPP_MEMORY_ALLOCATION_ERROR (C enumerator), 26
 NppStatus.NPP_MEMSET_ERROR (C enumerator), 24
 NppStatus.NPP_MIRROR_FLIP_ERROR (C enumerator), 26
 NppStatus.NPP_MISALIGNED_DST_ROI_WARNING (C enumerator), 27
 NppStatus.NPP_MOMENT_00_ZERO_ERROR (C enumerator), 26
 NppStatus.NPP_NO_ERROR (C enumerator), 27
 NppStatus.NPP_NO_MEMORY_ERROR (C enumerator), 26
 NppStatus.NPP_NO_OPERATION_WARNING (C enumerator), 27
 NppStatus.NPP_NOT_EVEN_STEP_ERROR (C enumerator), 24
 NppStatus.NPP_NOT_IMPLEMENTED_ERROR (C enumerator), 26
 NppStatus.NPP_NOT_SUFFICIENT_COMPUTE_CAPABILITY (C enumerator), 23
 NppStatus.NPP_NOT_SUPPORTED_MODE_ERROR (C enumerator), 26
 NppStatus.NPP_NULL_POINTER_ERROR (C enumerator), 26
 NppStatus.NPP_NUMBER_OF_CHANNELS_ERROR (C enumerator), 25
 NppStatus.NPP_OUT_OFF_RANGE_ERROR (C enumerator), 26
 NppStatus.NPP_OVERFLOW_ERROR (C enumerator), 24
 NppStatus.NPP_QUADRANGLE_ERROR (C enumerator), 25
 NppStatus.NPP_QUALITY_INDEX_ERROR (C enumerator), 24
 NppStatus.NPP_RANGE_ERROR (C enumerator), 26
 NppStatus.NPP_RECTANGLE_ERROR (C enumerator), 25
 NppStatus.NPP_RESIZE_FACTOR_ERROR (C enumerator), 25
 NppStatus.NPP_RESIZE_NO_OPERATION_ERROR (C enumerator), 24
 NppStatus.NPP_ROUND_MODE_NOT_SUPPORTED_ERROR (C enumerator), 24
 NppStatus.NPP_SCALE_RANGE_ERROR (C enumerator), 26
 NppStatus.NPP_SIZE_ERROR (C enumerator), 26
 NppStatus.NPP_STEP_ERROR (C enumerator), 26
 NppStatus.NPP_STREAM_CTX_ERROR (C enumerator), 24
 NppStatus.NPP_STRIDE_ERROR (C enumerator), 25
 NppStatus.NPP_SUCCESS (C enumerator), 27

NppStatus.NPP_TEXTURE_BIND_ERROR (C enumerator), 23
 NppStatus.NPP_THRESHOLD_ERROR (C enumerator), 26
 NppStatus.NPP_THRESHOLD_NEGATIVE_LEVEL_ERROR (C enumerator), 26
 NppStatus.NPP_WRONG_INTERSECTION_QUAD_WARNING (C enumerator), 27
 NppStatus.NPP_WRONG_INTERSECTION_ROI_ERROR (C enumerator), 24
 NppStatus.NPP_WRONG_INTERSECTION_ROI_WARNING (C enumerator), 27
 NppStatus.NPP_ZC_MODE_NOT_SUPPORTED_ERROR (C enumerator), 23
 NppStatus.NPP_ZERO_MASK_VALUE_ERROR (C enumerator), 25
 nppsThreshold_16s_Ctx (C function), 1852
 nppsThreshold_16s_I_Ctx (C function), 1852
 nppsThreshold_16sc_Ctx (C function), 1852
 nppsThreshold_16sc_I_Ctx (C function), 1853
 nppsThreshold_32f_Ctx (C function), 1853
 nppsThreshold_32f_I_Ctx (C function), 1853
 nppsThreshold_32fc_Ctx (C function), 1854
 nppsThreshold_32fc_I_Ctx (C function), 1854
 nppsThreshold_64f_Ctx (C function), 1855
 nppsThreshold_64f_I_Ctx (C function), 1855
 nppsThreshold_64fc_Ctx (C function), 1855
 nppsThreshold_64fc_I_Ctx (C function), 1856
 nppsThreshold_GT_16s_Ctx (C function), 1860
 nppsThreshold_GT_16s_I_Ctx (C function), 1860
 nppsThreshold_GT_16sc_Ctx (C function), 1860
 nppsThreshold_GT_16sc_I_Ctx (C function), 1861
 nppsThreshold_GT_32f_Ctx (C function), 1861
 nppsThreshold_GT_32f_I_Ctx (C function), 1861
 nppsThreshold_GT_32fc_Ctx (C function), 1861
 nppsThreshold_GT_32fc_I_Ctx (C function), 1862
 nppsThreshold_GT_64f_Ctx (C function), 1862
 nppsThreshold_GT_64f_I_Ctx (C function), 1862
 nppsThreshold_GT_64fc_Ctx (C function), 1863
 nppsThreshold_GT_64fc_I_Ctx (C function), 1863
 nppsThreshold_GTVa1_16s_Ctx (C function), 1867
 nppsThreshold_GTVa1_16s_I_Ctx (C function), 1868
 nppsThreshold_GTVa1_16sc_Ctx (C function), 1868
 nppsThreshold_GTVa1_32f_Ctx (C function), 1869
 nppsThreshold_GTVa1_32f_I_Ctx (C function), 1869
 nppsThreshold_GTVa1_32fc_Ctx (C function), 1870
 nppsThreshold_GTVa1_32fc_I_Ctx (C function), 1870
 nppsThreshold_GTVa1_64f_Ctx (C function), 1870
 nppsThreshold_GTVa1_64f_I_Ctx (C function), 1871
 nppsThreshold_GTVa1_64fc_Ctx (C function), 1871
 nppsThreshold_GTVa1_64fc_I_Ctx (C function), 1871
 nppsThreshold_LT_16s_Ctx (C function), 1856
 nppsThreshold_LT_16s_I_Ctx (C function), 1856
 nppsThreshold_LT_16sc_Ctx (C function), 1857
 nppsThreshold_LT_16sc_I_Ctx (C function), 1857
 nppsThreshold_LT_32f_Ctx (C function), 1857
 nppsThreshold_LT_32f_I_Ctx (C function), 1857
 nppsThreshold_LT_32fc_Ctx (C function), 1858
 nppsThreshold_LT_32fc_I_Ctx (C function), 1858
 nppsThreshold_LT_64f_Ctx (C function), 1858
 nppsThreshold_LT_64f_I_Ctx (C function), 1859
 nppsThreshold_LT_64fc_Ctx (C function), 1859
 nppsThreshold_LT_64fc_I_Ctx (C function), 1859
 nppsThreshold_LTVa1_16s_Ctx (C function), 1863
 nppsThreshold_LTVa1_16s_I_Ctx (C function), 1864
 nppsThreshold_LTVa1_16sc_Ctx (C function), 1864
 nppsThreshold_LTVa1_16sc_I_Ctx (C function), 1864
 nppsThreshold_LTVa1_32f_Ctx (C function), 1865
 nppsThreshold_LTVa1_32f_I_Ctx (C function), 1865
 nppsThreshold_LTVa1_32fc_Ctx (C function), 1865

nppsThreshold_LTVa1_32fc_I_Ctx (C function), 1866
 nppsThreshold_LTVa1_64f_Ctx (C function), 1866
 nppsThreshold_LTVa1_64f_I_Ctx (C function), 1866
 nppsThreshold_LTVa1_64fc_Ctx (C function), 1867
 nppsThreshold_LTVa1_64fc_I_Ctx (C function), 1867
 NppStreamContext (C struct), 47
 NppStreamContext.[anonymous] (C var), 49
 NppStreamContext.hStream (C var), 48
 NppStreamContext.hWorkspace (C var), 48
 NppStreamContext.nCudaDevAttrComputeCapabilityMajor (C var), 48
 NppStreamContext.nCudaDevAttrComputeCapabilityMinor (C var), 48
 NppStreamContext.nCudaDeviceId (C var), 48
 NppStreamContext.nMaxThreadsPerBlock (C var), 48
 NppStreamContext.nMaxThreadsPerMultiProcessor (C var), 48
 NppStreamContext.nMultiProcessorCount (C var), 48
 NppStreamContext.nReserved0 (C var), 48
 NppStreamContext.nSharedMemPerBlock (C var), 48
 NppStreamContext.nStreamFlags (C var), 48
 nppsXor_8u_Ctx (C function), 1833
 nppsXor_8u_I_Ctx (C function), 1834
 nppsXor_16u_Ctx (C function), 1833
 nppsXor_16u_I_Ctx (C function), 1834
 nppsXor_32u_Ctx (C function), 1833
 nppsXor_32u_I_Ctx (C function), 1834
 nppsXorC_8u_Ctx (C function), 1831
 nppsXorC_8u_I_Ctx (C function), 1832
 nppsXorC_16u_Ctx (C function), 1831
 nppsXorC_16u_I_Ctx (C function), 1832
 nppsXorC_32u_Ctx (C function), 1831
 nppsXorC_32u_I_Ctx (C function), 1832
 NppsZCType (C enum), 40
 NppsZCType.nppZCC (C enumerator), 40
 NppsZCType.nppZCR (C enumerator), 40
 NppsZCType.nppZCXor (C enumerator), 40
 nppsZero_8u_Ctx (C function), 1877
 nppsZero_16s_Ctx (C function), 1877
 nppsZero_16sc_Ctx (C function), 1878
 nppsZero_32f_Ctx (C function), 1878
 nppsZero_32fc_Ctx (C function), 1879
 nppsZero_32s_Ctx (C function), 1878
 nppsZero_32sc_Ctx (C function), 1878
 nppsZero_64f_Ctx (C function), 1879
 nppsZero_64fc_Ctx (C function), 1879
 nppsZero_64s_Ctx (C function), 1879
 nppsZero_64sc_Ctx (C function), 1879
 nppsZeroCrossing_16s32f_Ctx (C function), 1973
 nppsZeroCrossing_32f_Ctx (C function), 1973
 nppsZeroCrossingGetBufferSize_16s32f_Ctx (C function), 1973
 nppsZeroCrossingGetBufferSize_32f_Ctx (C function), 1973